NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DEVELOPMENT AND IMPLEMENTATION OF LOW-COST MOBILE SENSOR PLATFORMS WITHIN A WIRELESS SENSOR NETWORK

by

Michael Jay Tozzi

September 2010

Thesis Advisor:	Rachel Goshorn
Second Reader:	Duane Davis

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2010	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Development and Implementation of Low-Cost Mobile Sensor Platforms Within a Wireless Sensor Network	**5. FUNDING NUMBERS**
6. AUTHOR(S) Tozzi, Michael Jay	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ______N/A__________.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Sensor networks are used throughout the government and industry for a wide variety of purposes. Mobile Sensor Platforms (MSPs), from surface combatant vessels to unmanned aerial vehicles, have been integrated into these sensor networks since their inception. Unmanned MSPs currently used in sensor networks have two major drawbacks: They are extremely expensive and they require the control of a human operator. Remote controlled unmanned systems currently do not eliminate risk to personnel entirely, because they are typically too expensive to be considered expendable. If these standard unmanned systems are downed in a hostile environment, their recovery is often attempted by personnel on the ground; thus, still risking human lives.

The military is exploring the use of low-cost unmanned MSPs to eliminate the need to risk personnel in their recovery. One of the greatest expenses in the life cycle of any system is operator cost. To reduce or eliminate operator cost, a platform must be autonomous. Though algorithms exist for adding autonomous capabilities to a mobile platform, such algorithms are typically designed for robust systems with a great deal of processing power. Low-cost systems are typically limited in capability by a low-processing power CPU. For this reason, small footprint alternatives to existing autonomous control algorithms must be developed to truly implement a low-cost MSP.

This thesis applies the systems engineering process to developing a generic system solution for the need of a low-cost MSP, with concept of operations, external systems diagram, generic requirements, functional architecture and decompositions developed. The proposed generic system solution is then further designed in a scoped environment and implemented as a proof of concept prototype.

14. SUBJECT TERMS Systems Engineering, Intelligence Collection, Wireless Networks, Unmanned Vehicles	**15. NUMBER OF PAGES** 83
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DEVELOPMENT AND IMPLEMENTATION OF LOW-COST MOBILE SENSOR PLATFORMS WITHIN A WIRELESS SENSOR NETWORK

Michael Jay Tozzi
Lieutenant, United States Navy
B.S., United States Naval Academy, 2003

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2010

Author: Michael Jay Tozzi

Approved by: Rachel Goshorn
Thesis Advisor

Duane Davis
Second Reader

Clifford Whitcomb
Chairman, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Sensor networks are used throughout the government and industry for a wide variety of purposes. Mobile Sensor Platforms (MSPs), from surface combatant vessels to unmanned aerial vehicles, have been integrated into these sensor networks since their inception. Unmanned MSPs currently used in sensor networks have two major drawbacks: They are extremely expensive and they require the control of a human operator. Remote controlled unmanned systems currently do not eliminate risk to personnel entirely, because they are typically too expensive to be considered expendable. If these standard unmanned systems are downed in a hostile environment, their recovery is often attempted by personnel on the ground; thus, still risking human lives.

The military is exploring the use of low-cost unmanned MSPs to eliminate the need to risk personnel in their recovery. One of the greatest expenses in the life cycle of any system is operator cost. To reduce or eliminate operator cost, a platform must be autonomous. Though algorithms exist for adding autonomous capabilities to a mobile platform, such algorithms are typically designed for robust systems with a great deal of processing power. Low-cost systems are typically limited in capability by a low-processing power CPU. For this reason, small footprint alternatives to existing autonomous control algorithms must be developed to truly implement a low-cost MSP.

This thesis applies the systems engineering process to developing a generic system solution for the need of a low-cost MSP, with concept of operations, external systems diagram, generic requirements, functional architecture and decompositions developed. The proposed generic system solution is then further designed in a scoped environment and implemented as a proof of concept prototype.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AOA – Analysis of Alternatives

AOI – Area of Interest

CI – Configuration Item

COTS – Commercial Off The Shelf

CREW – Counter Radio-Controlled-Improvised-Explosive-Device Electronic Warfare

ESD – External Systems Diagram

IDEF0 – Integrated Definition for Function Modeling

JCA – Joint Capability Area

MSP – Mobile Sensor Platform

OPSIT – Operational Situation

OV1 – Operational View 1

UGV – Unmanned Ground Vehicle

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

While assigned to an Army unit on the ground in Iraq as Electronic Warfare Officer, the author of this thesis developed a method for using vehicle mounted CREW jamming systems' logging capability to collect signals intelligence data. This data proved invaluable in analyzing communication trends and predicting enemy attacks along routes transited by our troops. This, coupled with the high manpower requirements of remote controlled systems, the imminent force reduction in combat zones, and the risk of loss of human life in the attempted recovery of a disabled expensive system in a hazardous area identified a need for autonomous unmanned systems to be used as Mobile Sensor Platforms.

The author of this thesis followed the Systems Engineering "Vee" Model in developing a DRM to identify needs, concept of operations, external systems diagram, generic requirements, functional architecture and decompositions developed. The proposed generic system solution is then further designed in a scoped environment and implemented as a proof of concept prototype.

The DRM presented in this thesis identifies the Joint Capability Areas and FORCEnet missions covered by this system (see Figure 1 and Figure 2).

Tier 1	Tier 2A	Tier 2B
Joint Net Centric Operations	Information Transport	Information Transport
Joint Battlespace Awareness	Planning and Direction	Conduct Collection Management
	Observation and Collection	Radio Frequency
	Dissemination and Integration	Enable Smart Push/Pull for Intelligence Products

Figure 1. Joint Capability Areas from Naval Power 21 [From 10].

Mission Capability	Mission Sub-Capability
Communication and Networks/Infrastructure	Provide Information Transfer
Battlespace Awareness/ISR	Conduct Sensor Management and Information Processing
	Detect and ID Targets
	Provide Cueing and Target Information

Figure 2. FORCEnet Missions from Naval Power 21 [From 10].

The DRM also covers a number of requirements for the system. These missions and requirements were utilized in developing a functional architecture for a general solution system.

A functional architecture for a possible solution system was developed, with a snapshot of the functional architecture decomposition, first level, seen in Figure 3.

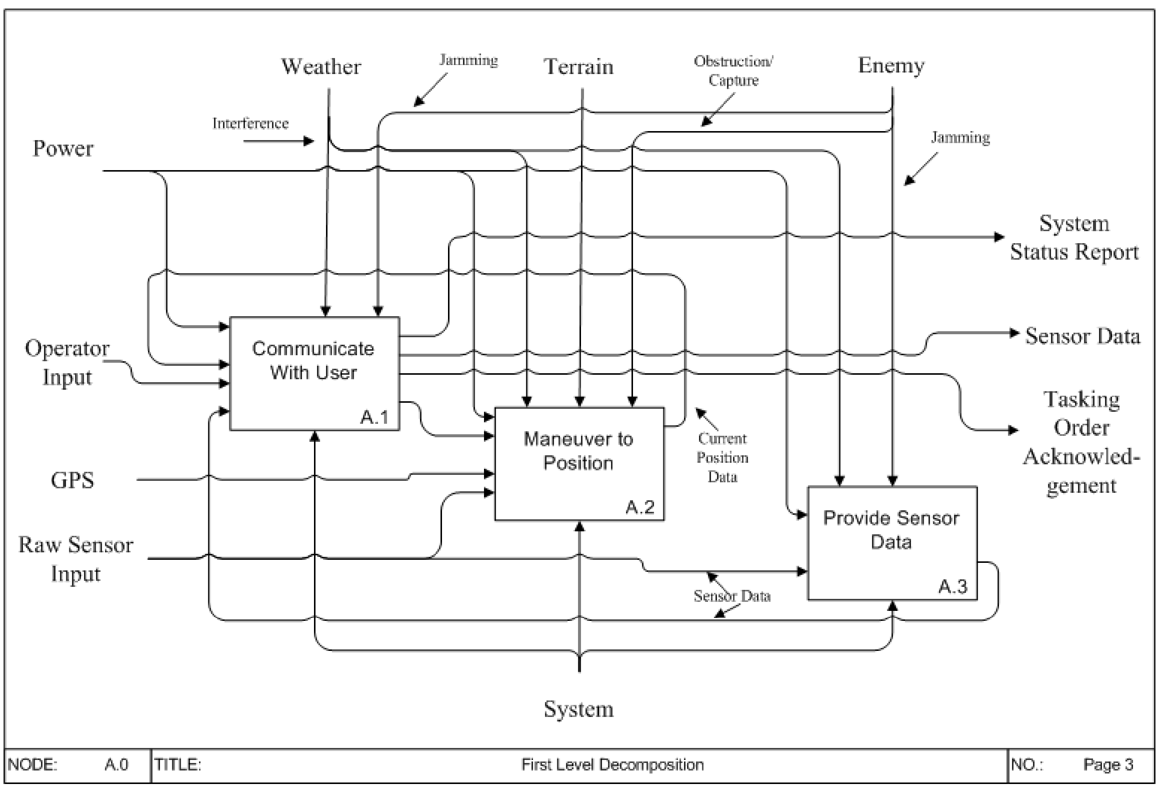

Figure 3. First Level Decomposition of Solution System.

Finally, the generic system solution was further designed and implemented in a scoped environment, as a proof of concept prototype. The system used was the Renegade SRV ground robot. This system was utilized because of its low cost (~$1200) and ready availability. The goal in developing the proof of concept prototype was to create small footprint algorithms for autonomous navigation for use on inexpensive robots. The motion tracking algorithm used in the proof of concept prototype was based on differential drive theory and utilized odometry values obtained from the wheel encoders of the robot as the robot transited.

The error rate of the odometry based motion tracking was successful in certain scenarios and in additional scenarios was determined to be too high for effective navigation. This was due to the limitations of the SRV robot's firmware in performing trigonometric calculations, and the error associated with wheel slip as the robot moves. The Renegade SRV ground robot is a first generation, low-cost robot and through this thesis, recommendations for updates can help improve this system and be used as expendable unmanned systems.

The author makes recommendations for areas of future research for both on the SRV robot and on other possible systems. Overall, there is a paradigm shift required in warfare, where low-cost, expendable, and automated unmanned systems are critical in the current and future Network-Centric Warfare.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author wishes to thank Professor Rachel Goshorn, Commander Duane Davis, and Professor Gary Langford for their guidance during the writing of this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This chapter will cover the overall purpose of this thesis. It includes a description of the problem, background on the subject, and a detailed table of contents.

A. PROBLEM STATEMENT

Mobile Sensor Platforms are not new to the U.S. military. A surface vessel with a commercial off-the-shelf (COTS) RADAR can be considered a mobile sensor platform. RADAR has been used by U.S. Naval, Ground, and Air forces for generations. Wireless communication has been used by the U.S. military for even longer [1]. Military forces have been relaying intelligence and other critical information over wireless systems for about a century [1]. Though the concept of Network Centric Warfare has only been formally stated in recent years, its roots can be traced back to the First World War. During the First World War, naval surface combatants could warn each other of enemy vessels spotted in the area by radio, flashing light, or signal flag [1]. These surface vessels could then use these same communication methods to coordinate an attack or other action deemed appropriate. This concept of long-distance communication between separate nodes for sharing of intelligence and coordination of actions is now so fundamental to the use of military force that one would be hard pressed to name a military action within the last 100 years in which it was not used.

Wireless communication and access to information have become so universal that professional military forces no longer have a monopoly over the use of network centric warfare. A group of insurgents or guerillas may now utilize mobile telephones or inexpensive 2-way radios, along with the Internet-based satellite imaging site “Google Earth” to coordinate an attack on U.S. forces [2]. In order to provide greater security for our forces, we must continually strive to be one step ahead of our adversaries. With the potential draw-down of troops in both theaters of operations, we must seek less manpower intensive solutions [3]. Unmanned systems are part of the solution, but current unmanned systems usually require direct operator control. Another limitation of current unmanned systems is stay time. Though other alternatives are under development, the vast majority of unmanned systems being utilized in military operations are Unmanned Aerial Vehicles. Though these systems have a proven track record for effective support of combat operations, they require fuel to stay on station. Even the most fuel efficient aircraft will need to land eventually to refuel. Lastly, simply automating unmanned systems does not reduce the risk to our personnel. The majority of our systems are so expensive that when downed they are not simply left behind. Soldiers are sent, often in hostile or otherwise dangerous territory, to recover these downed systems [4].

1. Personal Experience and Motivation

Prior to enrollment at the Naval Postgraduate School, I was assigned to the Joint Composite CREW (Counter Radio-Controlled-IED Electronic Warfare) Squadron 1 (JCCS-1) in Iraq. As a member of this Squadron, I was assigned to two separate Army commands, the 1-133 Infantry Regiment and the 2-504 Parachute Infantry Regiment, as the Electronic Warfare Officer (EWO). As the EWO, I was responsible for fielding and maintaining CREW jamming systems on ground vehicles operated by these units. These ground vehicles included High Mobility Multi-Purpose Wheeled Vehicles (HMMWVs) and M1117 Armored Security Vehicles (ASVs) as well as a number of other types of trucks. Many of the CREW systems included a data logging feature that recorded information on the electronic noise they encountered. The data included the frequency sensed by the system, the time, and the latitude and longitude of the vehicle at the time the signal was sensed. My team and I developed a method for using this data to provide signals intelligence on the area traversed by the soldiers of this unit. After a group returned from a mission, my team and I would sort the data by frequency and create a color-coded overlay in Falcon View (where Falcon View is a PC-based mapping application developed by the Georgia Tech Research Institute for the Department of Defense [5]). These color-coded overlays allowed us to determine trends in electronic activity in areas regularly transited by our troops. Any change in regular trends or other anomalies would be reported to the Battalion Commander and the battalion intelligence officer. For example: Urban areas normally exhibit a great deal of electronic noise related to wireless communication. A sudden reduction of electronic noise could indicate that the enemy is relocating residents in preparation for an attack. Conversely, open desert areas tend to have very little electronic noise. The sudden presence of electronic noise in an uninhabited area could indicate that enemy forces are planning an attack in the area. This type of signals intelligence, combined with tactics we developed for the use of CREW jamming systems, were used effectively by the paratroopers of the 2-504 PIR. On more than one occasion, enemy forces had attempted to use wireless systems to coordinate an ambush on our convoys. Our troops were prepared for these attacks, and they were able to defeat the enemy with zero casualties on our side. A number of awards were presented to my team and me for the methods and tactics we developed. Among these awards was the Bronze Star Medal, which was awarded to me.

The use of electronic sensors mounted in groups of ground vehicles for collection of signals intelligence was my inspiration for pursuing this thesis project. One major limitation of the methods used was that they required a mobile sensor platform to physically return from the area of interest (AOI) to provide the signals data. Another limitation was that the mobile sensor platforms required human operators for direct control. It is not practical for U.S. Forces to leave the above mentioned vehicles unattended in a hostile environment to collect signals intelligence data. For these reasons, it is my opinion that ground forces would benefit greatly from large numbers of expendable land-based mobile sensor platforms scattered around an AOI collecting signals intelligence data and transmitting it wirelessly to a distant operator or headquarters for analysis.

B. UNMANNED GROUND VEHICLES

The DoD has been developing Unmanned Ground Vehicles (UGVs) for over 20 years [6]. A number of UGVs have been fielded and tested in operational environments. Two of these systems stand out as possible candidates for integration into or basis of design for the solution to the problem stated above.

1. The XM1216 Small Unmanned Ground Vehicle (SUGV)

The XM1216 is a small, man-portable, lightweight UGV. It is designed to conduct military operations in urban and desert terrain, as well as inside tunnels, sewers and caves [7]. It can perform missions that in hazardous conditions without exposing soldiers to these risks directly.

Figure 4. XM1216 Small Unmanned Ground Vehicle (SUGV) [From 8]

The SUGV brings the following capabilities to units on the battlefield:

- Soldiers can use the SUGV to conduct extended reconnaissance of urban and complex terrain and subterranean areas [9].
- Provides vital information regarding buildings, field fortifications, tunnels, sewers, subways, bunkers, facilities, and other structures in support of military operations, peacekeeping, and other Stability and Reconstruction Operations (S&RO) [9].

- The Soldier will be able to conduct reconnaissance of a building, investigate suspected IED’s or send the SUGV into caves or tunnels to seek out the enemy. Sensor information can be transmitted over the network to all levels of battalion operations [9].
- The SUGV is an 80% scaled down version of the Packbot, of which hundreds have been fielded to support Operation Iraqi Freedom and Operation Enduring Freedom [9].
- The SUGV is a Soldier’s tool. The Soldier utilizes the Common Controller (CC) to send commands to the SUGV and receives imagery and other information from the SUGV [9].
- The SUGV can be used to clear buildings of suspected booby traps, inspect caves for weapon caches, search for IEDs, etc. [9].
- The SUGV platform weighs less than 32 lbs. and can be carried in a MOLLE pack. This is significantly lighter than current systems used in contingency operations in theater today [9].

Figure 5. SUGV being transported by a soldier on foot [From 8].

The SUGV is currently being evaluated by the Army Evaluation Task Force (AETF). It is scheduled to be fielded to all Brigade Combat Teams in the Army by 2025 [7].

Figure 6. ARMY soldiers deploying a SUGV during an operational test [From 8].

The SUGV provides an example of a small portable system that can be deployed by forces on the ground for providing sensor data on an AOI. Since it is modular, individual modules may be developed to tailor to the specific needs of the mission. Though the SUGV seems likely to make a fantastic candidate for a solution to the problem stated, it was not used for the proof of concept prototype due to time and budget constraints.

2. Autonomous Mobile Communication Relays

Law enforcement organizations and Special Operations forces operating in urban environments have reported problems in maintaining wireless communication links with robots that have entered buildings [10]. This is due to the fact that most radio communication systems used by these units are line-of-sight [10]. To address this problem, the Space and Naval Warfare Systems Command (SPAWAR) has developed a number of Autonomous Mobile Communication Relays (AMCRs) [10].

Figure 7. AMCR “Master” robot followed by “slave” communication relay robots [From 10].

In the AMCR project, a “Master” robot would enter a building and deploy a convoy of less sophisticated and expensive “slave” robots. These “slave” robots serve as communication relays for the “master” robot. They also utilize onboard sensors to ensure that a space that has been previously cleared of enemy personnel or other hazards by the “master” robot remains cleared [10].

Though the AMCR program has since transitioned from a research project to project of specific practical application [10], the concept of the use of small, inexpensive autonomous robots with sensors and communications capability may be applied to a possible solution for the military problem stated above. For example, the “Master” robot could be a mobile Command and Control (C2) Center, from and to which the squad of low-cost and expendable unmanned systems could communicate; from the mobile C2 Center, situational awareness of the AOI could be wirelessly communicated back to a C2 Headquarters.

C. SYSTEMS ENGINEERING OVERVIEW

Using a Systems Engineering approach, this thesis proposes a generic solution for the problem of providing sensor data and signals intelligence, on an overly large or hazardous AOI to a ground force of reduced size, by first presenting the initial concept, an external systems diagram, the system requirements, and a generic functional architecture (hierarchy and decompositions). The thesis then presents a proposed specific solution to the problem, as well as research into a specific proof of concept prototype. This thesis, therefore, applies the entire Systems Engineering “Vee” (described in Chapter II) to the critical military need. As discussed later, this thesis will apply the left-hand side of the “Vee” to a generic solution, and the right-hand side of the “Vee” to implementation and verification in a scoped proof of concept system. The thesis then suggests areas of further research to find a specific solution to the military problem stated above.

This thesis applies the Systems Engineering Process to address the capability gap of providing Joint Battlespace Awareness (addressed in Chapter III) on an AOI. A Design Reference Mission (DRM) is presented in Chapter III, which addresses the specific problem in detail. The details of the operating environment are addressed. Architecture for a generic solution to the problem is presented. A specific solution is proposed and a proof of concept prototype is developed.

D. THESIS OUTLINE

This section provides an overview of the chapters of the thesis, as well as a brief description of the contents of each chapter section. Each chapter applies the System Engineering process by building upon the previous chapter.

1. Chapter II: Application of the Systems Engineering Process

This chapter covers the specific Systems Engineering Process, the "Vee" model, that was applied to the military problem. It describes the steps taken to develop the generic system architecture, the specific proposed solution, as well as the steps taken towards the development of the proof of concept prototype.

2. Chapter III: The Design Reference Mission

This chapter presents the Design Reference Mission (DRM) for the problem. The DRM includes the following:

- Problem Definition
- Operational Need
- Operational Situation (OPSIT) Generation
- Projected Operating Environment (POE)
- Threat
- Assumed Threat General Conditions
- Metrics
- Mission Success Requirements
- Mission Definition
- Operational Activities

- Operational Tasks
- Mission Execution
- Operational Concept

Overall, the DRM helps to bound the problem and provide guidance for what the system must accomplish for successful completion of the intended missions of the system.

3. Chapter IV: Generic System Architecture

This chapter provides the architecture for a generic solution to the problem. It presents an External Systems Diagram (ESD), generic system requirements, and provides a functional hierarchy and a functional decompositions of the system. Each level of the Functional Architecture is decomposed and presented in an IDEF0 diagram. The ESD, requirements, functional hierarchy, and functional decomposition are derived from the DRM in the previous chapter.

4. Chapter V: Proof of Concept Prototype

This chapter presents research into a specific proof of concept prototype developed as a possible future solution to the problem. Concepts related to robotic navigation are described. The Surveyor SRV robot which was used is presented in detail. The Surveyor SRV robots are a new part of the Naval Postgraduate School's Network-Centric Systems Engineering Track, of which this thesis is a part of. Results of the Network-Centric Systems Engineering Laboratory research, for the proof of concept system, related to the Surveyor SRV are provided in this chapter.

5. Chapter VI: Summary, Recommendations, and Conclusion

This chapter summarizes the previous chapters of the thesis. In addition, it provides recommendations for future research, as well as a conclusion for the systems engineering research related to the project.

THIS PAGE INTENTIONALLY LEFT BLANK

II. APPLICATION OF THE SYSTEMS ENGINEERING PROCESS

This chapter presents the Systems Engineering (SE) process that was utilized to develop a solution to the problem stated in Chapter I. It provides a brief overview of how the SE "Vee" process was applied to create the architecture of a generic solution and how this generic architecture was applied to build a proof of concept prototype.

A. SYSTEMS ENGINEERING PROCESS

Systems Engineering is often defined as a multidisciplinary engineering discipline in which decisions and designs are based on their effect on the system as a whole [11]. Formal methods have been designed to establish system requirements and aid in the development and decision making processes. The goal in the case of this thesis is to design a system that would eventually meet the needs of forces requiring intelligence on an AOI. This thesis provides the concept, external systems diagram, generic requirements, and functional architecture of the system needed to solve the problem. The architecture is then applied to develop a proof of concept prototype. In this case, the prototype is the Surveyor SRV robot.

B. SYSTEMS ENGINEERING V-MODEL

Though there are many models for implementing the Systems Engineering process, the "Vee" model, shown in Figure 8, was applied in this thesis.

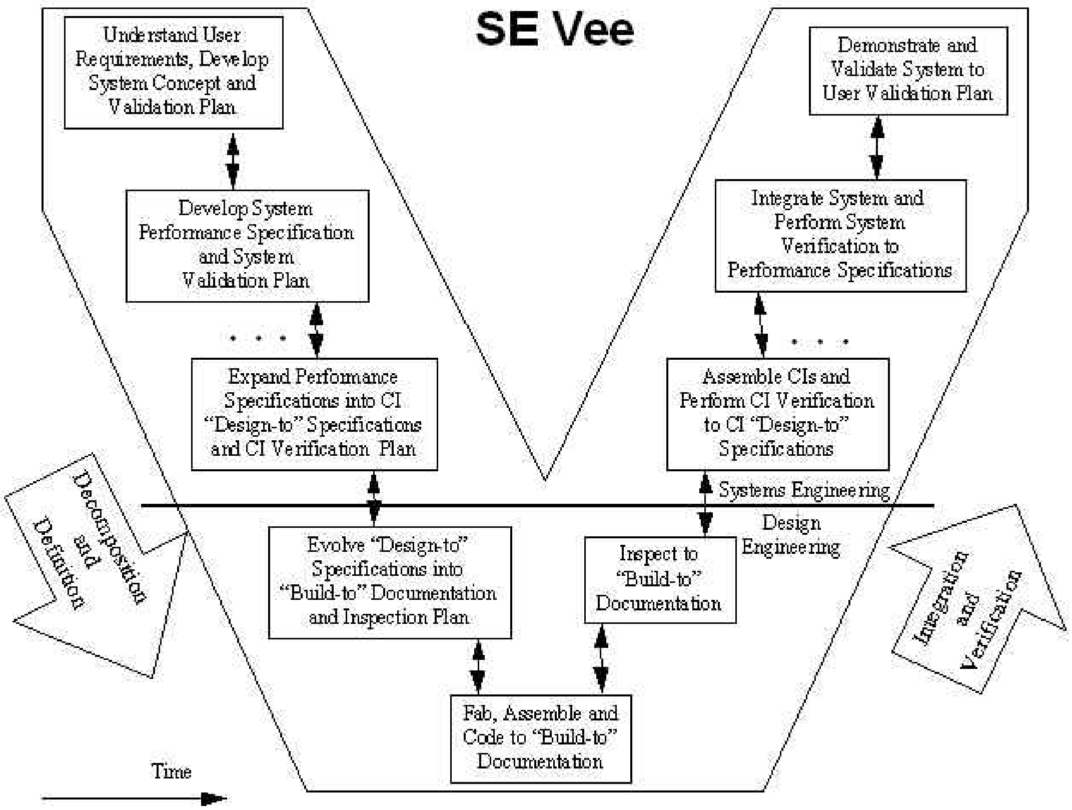

Figure 8. The Systems Engineering "Vee" Model [11]

In utilizing the "Vee" model, the design starts at the upper left corner with "Understand User Requirements, Develop System Concept and Validation Plan." The needs of the user were stated in the problem statement. These were applied in the development of the DRM (covered in Chapter III of this thesis). The DRM formally states the Requirements, System Concept, and Validation Plan.

The next step in the application of the "Vee" Model is the Development of Configuration Items (CIs) and a System Validation Plan. The CIs are developed in the Generic System Architecture (covered in Chapter IV). The System Validation Plan is presented in the application of the Generic System Architecture in development of the Proof of Concept Prototype (covered in Chapter V).

The next few steps along the bottom left-hand corner, through the first part of the right hand corner, of the "Vee" Model include developing specifications, building components to these specifications, and testing components as they are built. These steps in the process were covered in the development of the Proof of Concept Prototype (covered in Chapter V).

This thesis does not cover the upper right-hand corner of the “Vee” Model, which involves integrating the system components onto the SRV robot and performing a final validation test. This is because the final system was not completed, due to constraints of time and funding. This will be explained further in Chapter V and Chapter VI.

In summary, this chapter covered the systems engineering process that was applied in this thesis. It covered the specific SE model used and how it was applied in developing a solution to the problem. The systems engineering products developed are shown in the following chapters.

THIS PAGE INTENTIONALLY LEFT BLANK

III. DESIGN REFERENCE MISSION

This chapter covers the Design Reference Mission (DRM). The DRM formally states the problem and requirements of the system and links these requirements to Joint and Navy capability requirements. These requirements are used as the basis of the Generic System Architecture.

A. DESIGN REFERENCE MISSION

1. Problem Statement

As discussed in Chapter I, as the number of troops on the ground are decreased, the need for Mobile Sensor Platforms (MSPs) to provide intelligence on the AOI will continue to increase. With fewer troops available to provide direct control for MSPs, an automated solution will become critical. Current autonomous MSPs tend to be so expensive; they are too expensive to be considered expendable. When they fail in hazardous areas, troops are placed at risk in attempting their recovery. For these reasons, an autonomous MSP that is inexpensive enough to be abandoned, if it fails in a hazardous area, is needed.

2. Operational Need

One of the greatest expenses in the life cycle of any system is operator cost. To reduce, or eliminate operator cost, a platform must be able to execute tasking without being controlled directly by an operator. U.S. Forces need an inexpensive system that can receive specific tasking on collection and transmission of intelligence from an AOI, execute this tasking autonomously, and await follow-on tasking.

3. Operational Situation (OPSIT) Generation

Operational Situations (OPSITs) are discrete multi-engagement events with specified operational characteristics [11]. Operational situations can be determined by defining the operating environment of the system and making a number of assumptions. The assumptions should include factors such as enemy strength and capabilities, weather conditions of the operating area, and location of supporting friendly forces. OPSITs will be defined in the next few sections, starting with Projected Operating Environment.

4. Projected Operating Environment (POE)

Though MSPs are being developed for the sea and the air as well as the land, this thesis will focus on the MSPs operating on the land. Ground-based MSPs, in the form of UGVs, will have to operate in the same environments as manned ground vehicles. They should be able to handle moderately rough terrain and inclement weather, such as rain and heavy winds. They should also be able to ford wet lands such as swamps.

5. Threat

UGVs may operate in an area with enemy forces present. Since we are interested in creating expendable systems, our concern for the threat is not in protecting people but in prolonging the effective life of the system.

6. Assumed Threat General Condition

While the MSP is in the Area of interest, it may be detected by an enemy unit. The enemy unit will attempt to disable the MSP by destroying it with force or capturing it.

7. Metrics

A number of metrics will be used to determine if the system can meet the requirements of the given scenario. These metrics are based on requirements in the Navy Tactical Task List [13] and the Universal Joint Task List [14]. The metrics stated in Table 1 will be used to map requirements to physical properties of the system and aid in selecting the best solution to the problem.

Metric #	Metric Type	Description of Metric	Supporting Document
M1	Seconds	Time to maneuver to location	NTA 2.2 Collect data and intelligence
M2	Millimeters	Distance from ordered location (error)	NTA 2.2 Collect data and intelligence
M3	Seconds	Time to transmit data collected from ordered location	NTA 2.2 Collect data and intelligence

Table 1. List of Metrics [From 11].

8. Mission Success Requirements

Mission success requirements are based on the stated functional requirements of the system. All requirements must be met for the mission to be considered successful. The activities required for mission success are divided into the following categories:

- Manage propulsion
- Maneuver to Location
- Receive Orders
- Collect Data
- Transmit Data
- Give System Status Report

9. Mission Definition

There are two categories of missions that the system must complete. The first category is made up of Joint Capability Areas (JCAs). The JCAs are taken from the Naval Power 21, which is a combination of Sea Power 21 and Expeditionary Maneuver Warfare Capabilities. Naval Power 21 has five pillars, which are Sea Shield, Sea Strike, Sea Basing, Expeditionary Maneuver Warfare, and FORCEnet [11]. The second category of missions are FORCEnet mission capabilities. They are also from Naval Power 21 [11].

Tier 1	Tier 2A	Tier 2B
Joint Net Centric Operations	Information Transport	Information Transport
Joint Battlespace Awareness	Planning and Direction	Conduct Collection Management
	Observation and Collection	Radio Frequency
	Dissemination and Integration	Enable Smart Push/Pull for Intelligence Products

Table 2. Joint Capability Areas from Naval Power 21 [From 11].

Mission Capability	Mission Sub-Capability
Communication and Networks/Infrastructure	Provide Information Transfer
Battlespace Awareness/ISR	Conduct Sensor Management and Information Processing
	Detect and ID Targets
	Provide Cueing and Target Information

Table 3. FORCEnet missions from Naval Power 21 [From 11].

10. Operational Activities

In order to accomplish each of the given missions, the system must be able to perform a number of tasks. These tasks are taken from the Common Operational Activities List (COAL) [15]. The COAL was used because it provides a standard list for all services to use in the specification of operational activities. The required operational activities of the system are as follows:

- Monitor the Area of Interest (AOI) (2.0 ID 612)
- Manage sensors and information processing (2.0 ID 459)
- Understand the situation (2.0 ID 950)
- Recognize threats (2.0 ID 951)
- Observe and Collect (2.0 ID 519)

- Task Sensor (2.0 ID 522)
- Control Sensor (2.0 ID 525)
- Collect and Transport Sensor Derived Data (2.0 ID 530)
- Collect Data (2.0 ID 544)
- Collect Contact Data (2.0 ID 545)
- Find Target of Interest (2.0 ID 613)
- Identify/Recognize Target of Interest (2.0 ID 614)

Operational Activities represent actions that the system must perform to accomplish a given mission. One example is in the case of the Operational Activity "Collect Data." In this case, the system must be able to utilize sensors to collect data on the AOI. This data may be visual or some other form of electromagnetic radiation.

11. Operational Tasks

The above Operational Activities were used to determine the Operational Tasks of the system. Operational Tasks provide guidance to the system on how to perform the individual Operational Activities. The specific Operational Tasks are critical in determining the requirements for successful completion of a given mission. The following Operational Tasks are taken from the Naval Tactical Task List (NTTL) and the Universal Joint Task List (UJTL).

- Communicate Information (NTA 5.1.1) [13]
- Conduct Collection Planning and Directing (NTA 2.1.3) [13]
- Collect Target Information (NTA 2.2.1) [13]
- Perform Tactical Reconnaissance (NTA 2.2.3.2) [13]

The Operational Activities and Operational Tasks listed above are used to determine the functional requirements and develop a functional architecture of the system.

12. Mission Execution

Successful mission execution depends on successful completion of the Operational Activities and Operational Tasks. Two specific missions for the system are:

- Monitor an Area of Interest and provide sensor data
- Track specific targets in an Area of Interest

13. Operational Concept

The Operational Concept provides a broad overview of what a system is and how it will be used. It should provide a set of scenarios that specifically demonstrate how the system will be used based on its interactions with external systems or users [11]. In the case of the system of interest, the following scenarios provide the operational concept:

- An operational command needs to collect data on a specific Area of Interest that may be dangerous for a manned system to enter. The command will deploy one or more of the MSPs at or near the AOI. These sensors will maneuver around a location ordered by an operator, collect sensor data, and transmit this data back to headquarters. The system will continue to follow preset instructions until it receives follow-on tasking.
- An operational command needs to deploy an ad-hoc sensor network in a hostile area. The command will draft a set of instructions for one or more systems to stand by in a specific location, collect sensor data, and transmit this data back to headquarters. These orders may contain instructions for each system to patrol a set route or patrol randomly within a bounded area. The command will deploy the system(s) with these orders. The system(s) will follow the instructions in the orders and transmit sensor data on the AOI back to headquarters.

In either of the above scenarios, the following conditions apply:

- Upon mission completion, if a system is able to return to the deploying command, it will make an effort to do so.
- If the system is unable to return (due to damage, capture, or obstacles), or if the deploying command determines it is unsafe for the system to return to headquarters (i.e., the system has become contaminated with a Nuclear, Biological, or Chemical agent) the system will not make an effort to return to the deploying command. Since the system is inexpensive and contains no sensitive data or components, it is considered expendable and no recovery effort will be made. Expendable robots could also be made to self-destruct to destroy hazards such as IEDs.

The above scenarios will help to bound the problem to the specific functions required in order for the system to accomplish the above stated missions. The DRM above will be used to create the architecture and requirements for the generic solution. The DRM will also serve as a set of guidelines in the production and testing of the proof of concept prototype.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. GENERIC SYSTEM ARCHITECTURE

The generic system architecture provides a set of guidelines for creation of a system that meets the needs outlined in the DRM. It makes up the center of the left side of the "Vee" Model. The generic system architecture is made up of the Operational View 1, the external systems diagram, the requirements of the system, and the functional architecture of the system.

A. OPERATIONAL VIEW 1 (OV1)

The Operational View 1 (OV1) helps give a broad "big picture" vision of what the system is and what it is supposed to do. The OV1 for the solution system is presented in Figure 9.

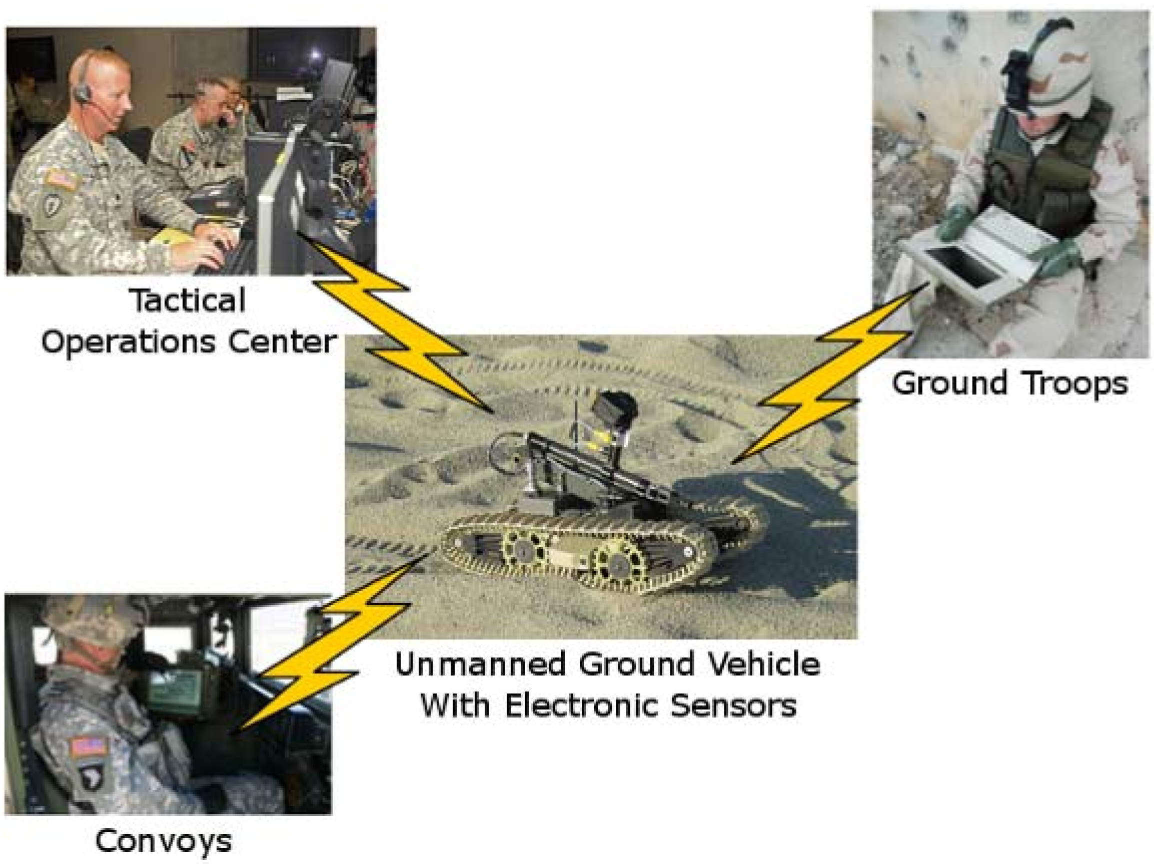

Figure 9. OV1 for the system solution [After 16, 17, 18].

The OV1 demonstrates how the solution system will be utilized to improve situational awareness and aid in collection of intelligence. The system pictured is the PACKbot UGV mentioned in Chapter I. It is not proposed as a specific solution but is used to give a conceptual view of what a solution might look like. In the OV1 figure, the UGV is communicating wirelessly with soldiers on foot, soldiers mounted in vehicles on convoy, and soldiers analyzing data at a Tactical Operations Center (TOC). In all cases, the UGV is providing intelligence and other sensor data to aid in tactical decision making.

B. EXTERNAL SYSTEMS DIAGRAM

The Integrated Definition for Function Modeling (IDEF0) format was used to develop the generic system architecture. The IDEF0 format consists of an External Systems Diagram (ESD) and a series of IDEF diagrams, labeled IDEF0-IDEF6.

The External Systems Diagram (ESD) serves to display the external systems that interact with the system, and bounds the design space (i.e., what is the system to design, its interfaces, and what are the external systems it must interface with, and system constraints, inputs and outputs?). At the top of the diagram are constraints. At the bottom of the diagram are the system and external systems. Functions are represented by the blocks in the center of the diagram. Each block has the top-level function of the corresponding system. Inputs, to system functions, point from the left into the function. Outputs from the function point to the right out of each related function.

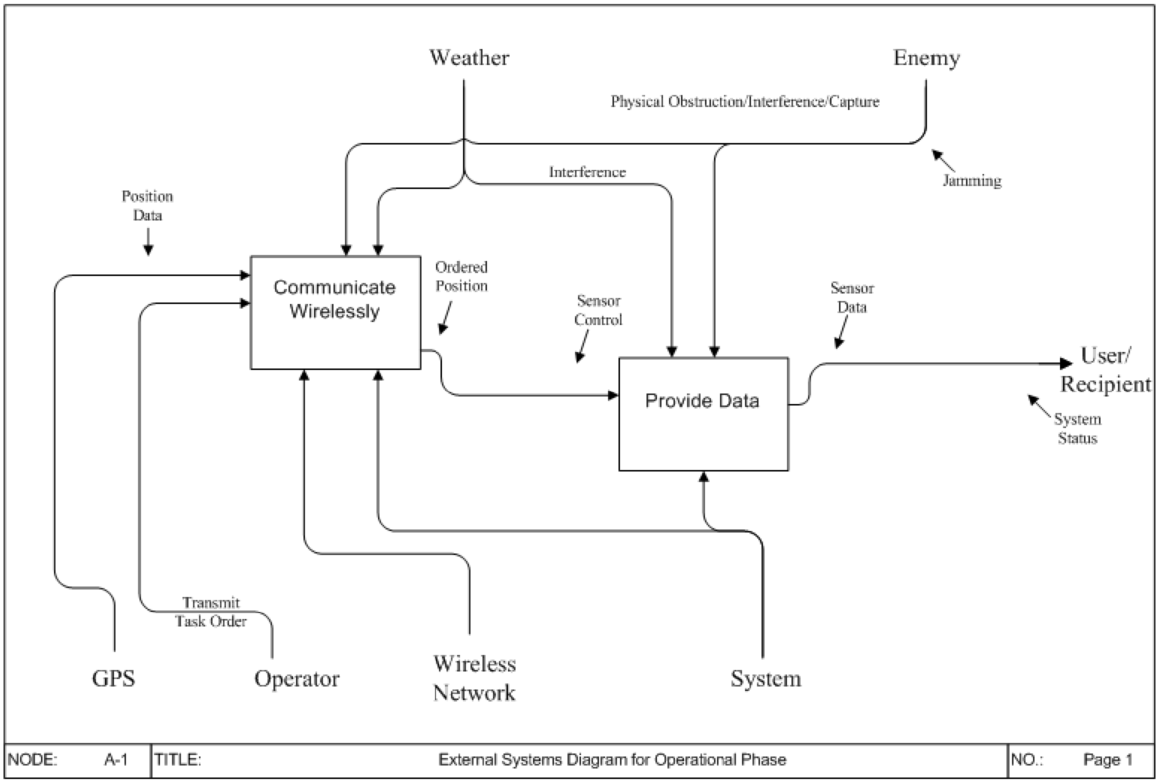

Figure 10. External Systems Diagram for Solution System.

As demonstrated in Figure 10, the system will primarily have to interact with two major external systems: The wireless network, which allows the system to communicate with the operator, and GPS, which provides navigation data.

IDEF diagrams serve as functional model for processes within a system. The IDEF model is used because it:

- Answers definitive questions about the transformation of inputs into outputs by the system [12].
- Establishes the boundary of the system on the context page. This boundary is explicated, if needed, as a meta description [12].

- Has one viewpoint; the viewpoint is the vantage or perspective from which the system is observed [12].
- Is a coordinated set of diagrams, using both a graphical language and a natural language [12].

C. REQUIREMENTS

Ideally, a set of requirements for a system would be determined by meeting with all stakeholders of the system. Such an elaborate undertaking would be prohibitively expensive in terms of time and money and is thus beyond the scope of this project. The requirements for the generic system were derived from the Concept of Operations (CONOPS) outlined in the DRM and the External Systems Diagram presented above.

Requirements

A.1.0—Input/output requirements

A.1.1—Input requirements

A.1.1.1— The system shall receive power from an installed power source.

A.1.1.2— The system shall receive raw data from installed sensors.

A.1.1.3— The system shall receive tasking from the user.

A.1.1.4— The system shall receive GPS position data.

A.1.1.5— The system shall receive physical support from the medium (ex. Ground for land based sensors, Air for Aerial vehicles, Ocean for Surface vessels).

A.1.2—Output requirements

A.1.2.1— The system shall provide acknowledgement of tasking receipt to the user.

A.1.2.2 — The system shall provide sensor data from installed sensors to the user.

A.1.2.3— The system shall provide reports on its own status to the user.

A.2.0—External systems requirements

A.2.1—The system shall interface with the user via wireless network.

A.2.2— The system shall receive position data from GPS satellites.

A.2.3— The system shall receive physical support from the medium (ex. Ground for land based sensors, Air for Aerial vehicles, Ocean for Surface vessels).

A.3.0—System constraint requirements

A.3.1— The system is constrained by terrain (medium, ex. Air for aerial vehicles, ocean for surface vessels).

A.3.2— The system is constrained by interference provided by weather.

A.3.3— The system s constrained by jamming and physical obstruction by the enemy.

A.4.0—The system requirements

A.4.1— The system shall provide situational awareness on the AOI by use of on-board sensors.

A.4.2— The system shall provide ability to re-task the system in the event of mission change.

A.4.3 — The system shall be inexpensive enough to be considered expendable.

D. FUNCTIONAL ARCHITECTURE

The functional architecture of a system contains a hierarchical model of the functions performed by the generic system and a functional architecture decomposition [12]. The functions within the functional architecture are derived from the requirements of the system. Ideally all functions will map to requirements.

Functional Decomposition

The following is the functional decomposition of the Generic solution for the problem.

A.0 Provide Operational Awareness of the AOI

A.1.0 Communicate with User

A.1.1 Receive Data from User

A.1.2 Transmit Data to User

A.1.2.1Transmit Data Receipt Acknowledgement to User

A.1.2.2Transmit Sensor Data to User

A.1.2.3Transmit Status Report to User

A.2.0 Maneuver to Position

A.2.1 Determine Current Pose

A.2.2 Determine Bearing and Range to Ordered Pose

A.2.3 Utilize Propulsion to Maneuver to Destination

A.2.4 Update Current Pose Data

A.3.0 Provide Sensor Data

A.3.1 Collect Data

A.3.1.1 Receive Sensor Data

A.3.1.2 Perform Diagnostic Self-Test

A.3.2 Store Data

The functional hierarchy of the system is better represented with the following functional hierarchy diagram (Figure 11).

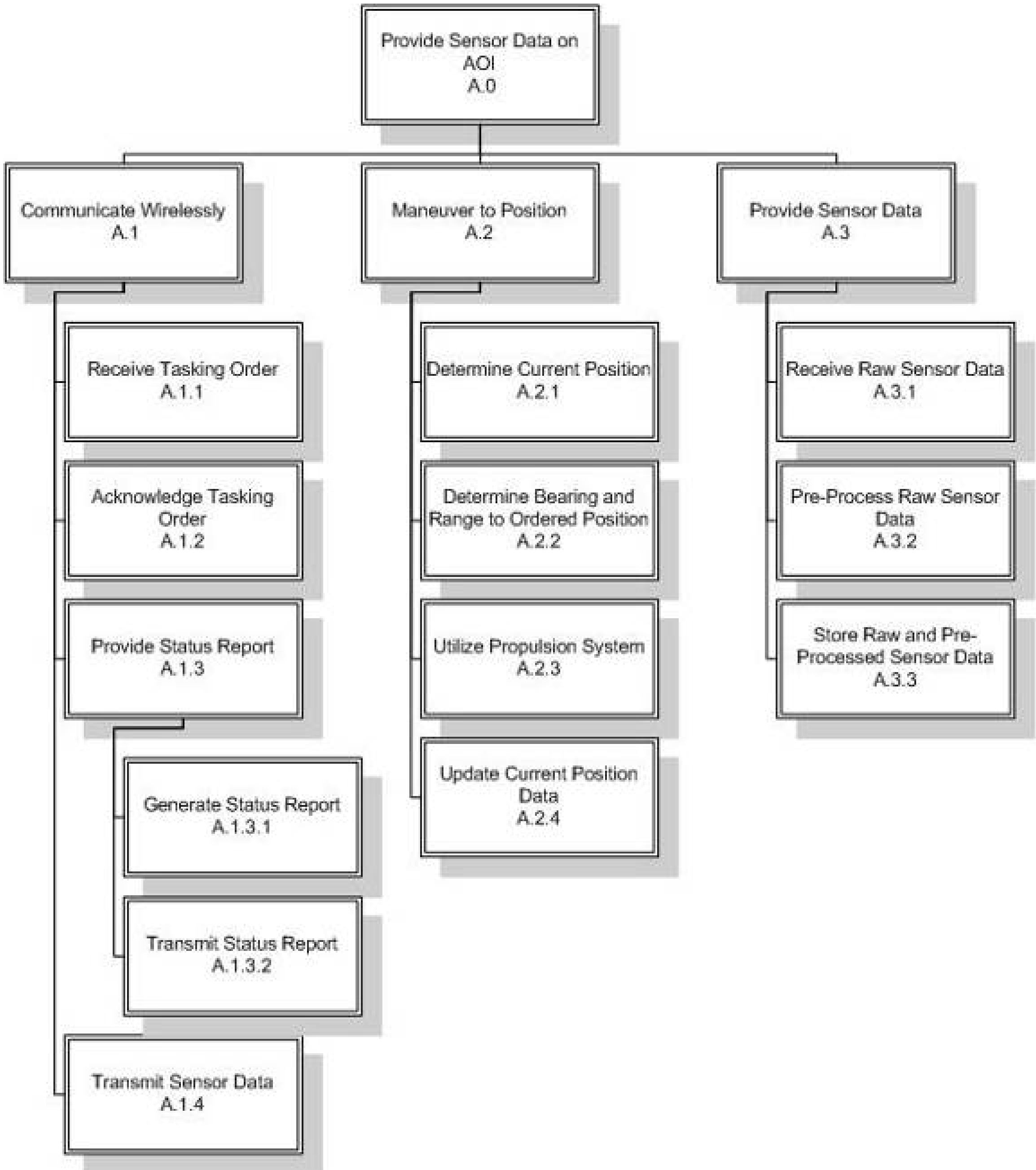

Figure 11. Functional Hierarchy Diagram for Solution System.

According to the functional hierarchy, three major functions must be performed in order to perform the function of "Provide Operational Awareness of the AOI."

- Communicate with User
- Maneuver to Position
- Provide Sensor Data

The IDEF0 model is used to decompose the functional hierarchy into component functions. The top level function is presented below. In keeping with the IDEF0 model, inputs, output, users, and constraints are represented as they were in the External Systems Diagram above (see Figure 10).

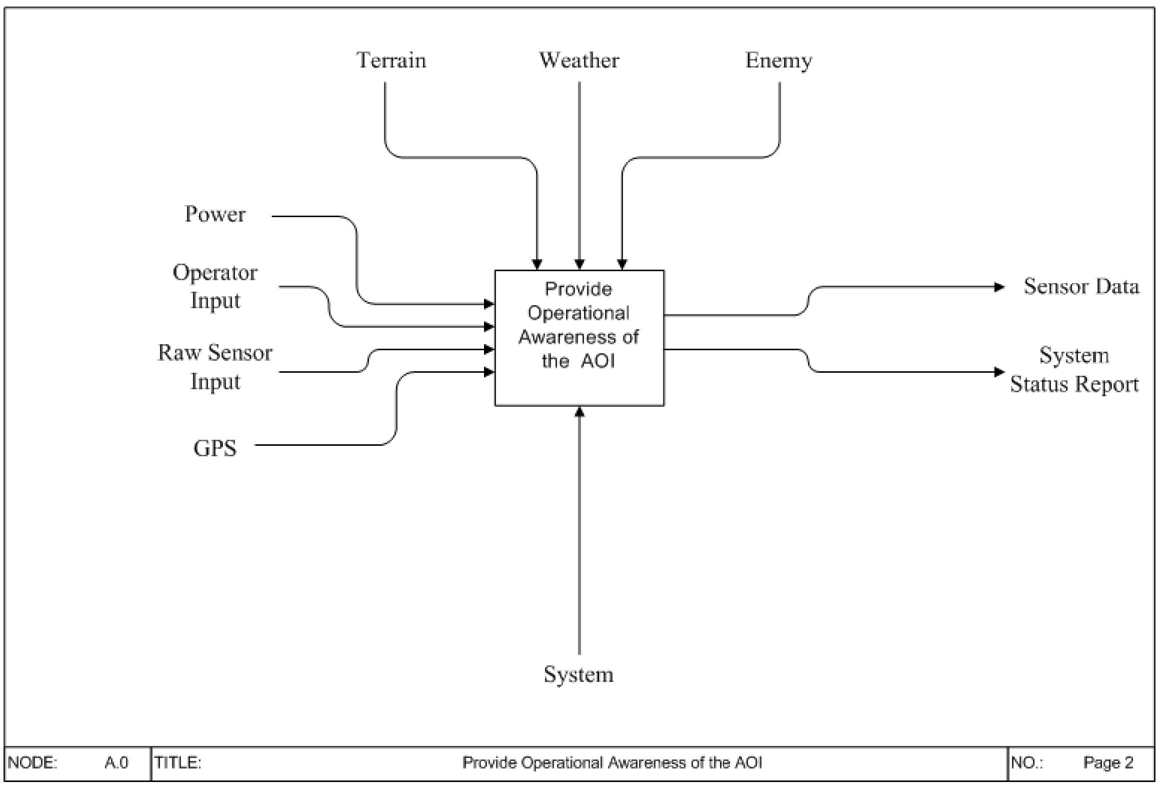

Figure 12. Top Level Function Diagram for Solution System.

The top level function has been decomposed into component functions and is presented in Figure 13.

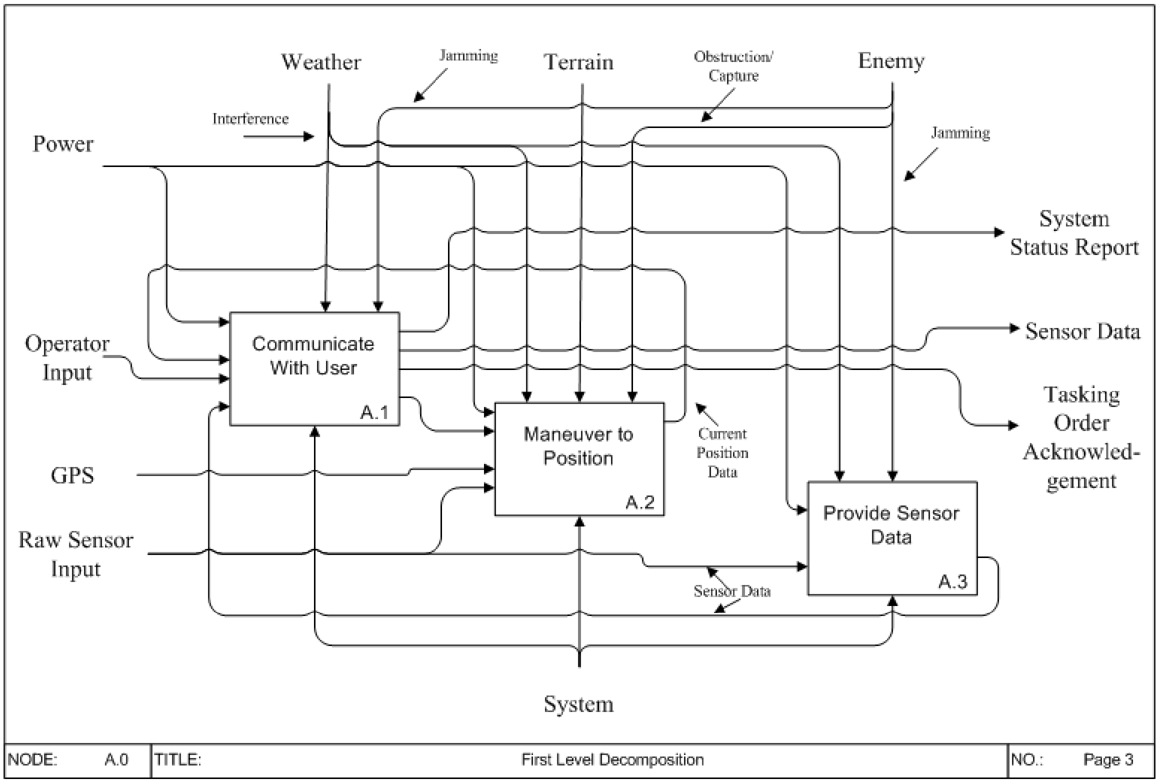

Figure 13. First Level Functional Decomposition.

The function "Communicate With User" has been decomposed and presented below. According to this decomposition, the system must be able to receive data from and transmit data to the user. This data transmitted to the user will include acknowledge of receipt of data from the user, data from the system's on-board sensors, and periodic status reports of the system itself. The subfunctions are constrained.

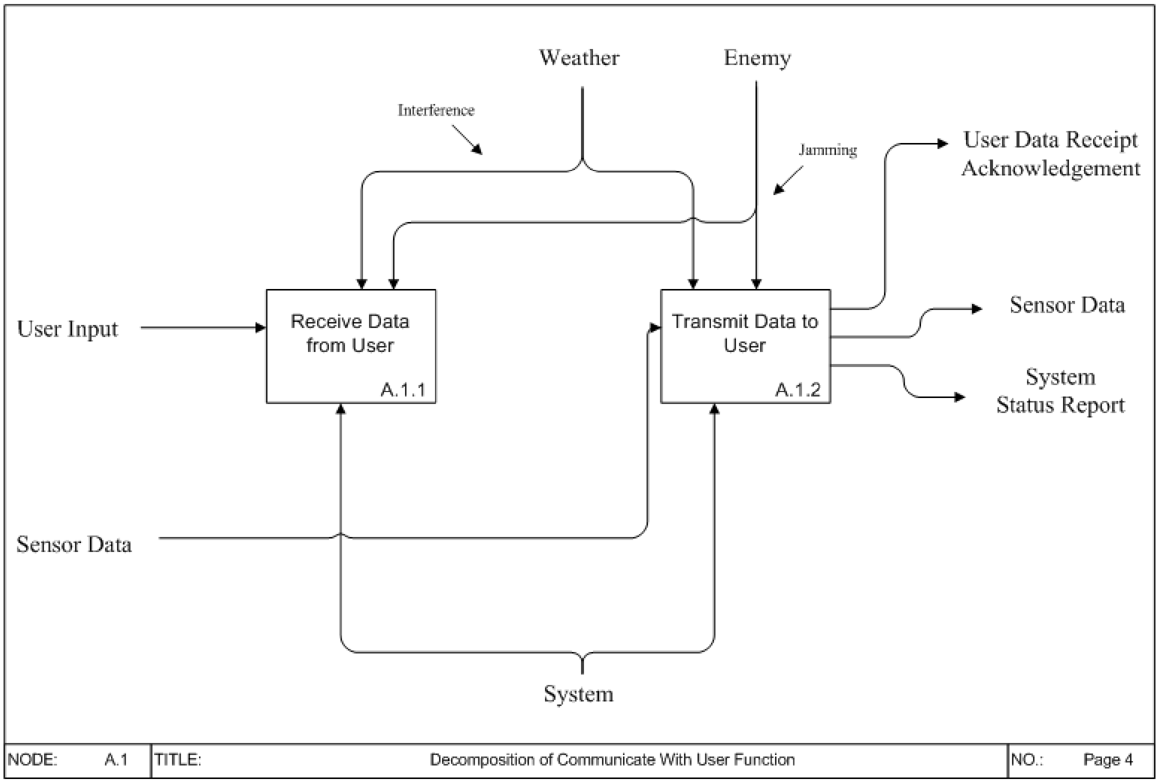

Figure 14. Decomposition of "Communicate With User" function.

The function "Transmit Data to User" can be decomposed further. In order to perform the parent function, it must perform the subfunctions of "Transmit Data Receipt Acknowledgement to User," "Transmit Sensor Data to User," and "Transmit Status Report to User." Like the parent function, these subfunctions are also constrained by interference from weather and jamming from the enemy.

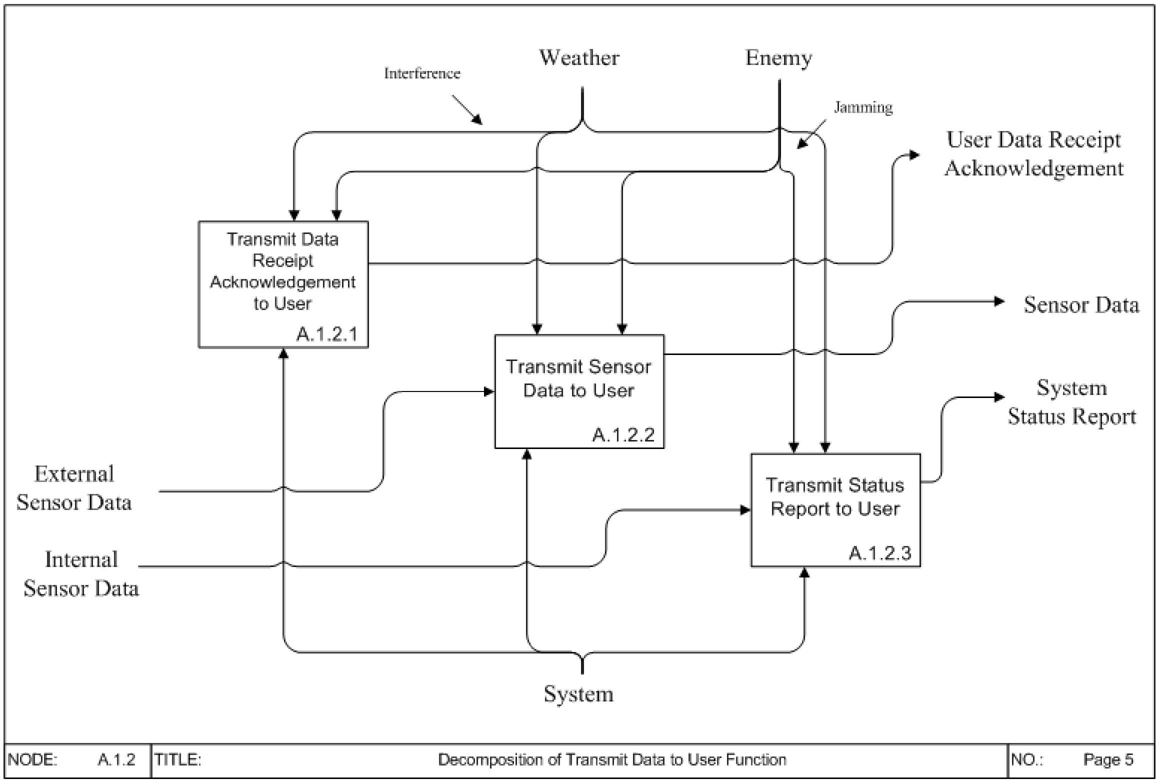

Figure 15. Decomposition of "Transmit Data to User" function.

The "Maneuver to Position" function is made up of a number of subfunctions. The "Pose" of the system is defined as the location and orientation (bearing) of the system. In order to accomplish the parent function, the system must first determine its current pose, calculate the bearing and range to the ordered pose, utilize its propulsion system to arrive at that pose, and store the new pose in memory.

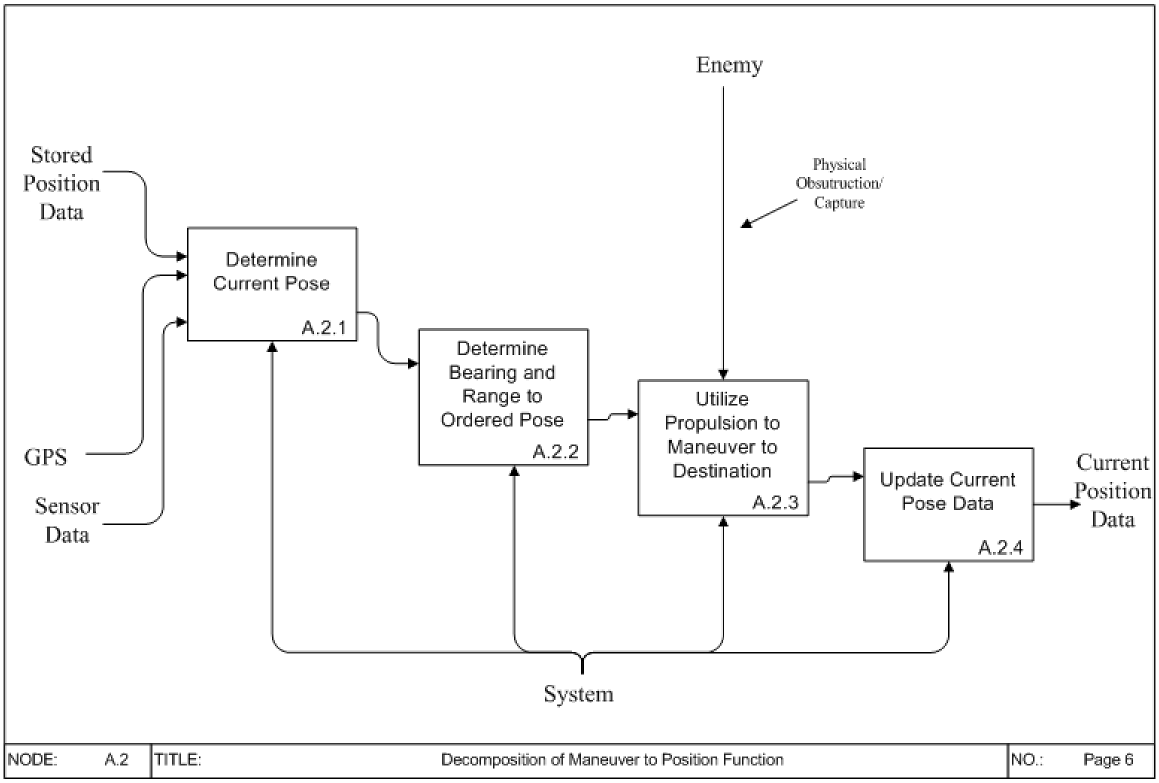

Figure 16. Decomposition of “Maneuver to Position” function.

In order to perform the function “Provide Sensor Data,” the system must perform the subfunctions of “Collect Data” and “Store Data.”

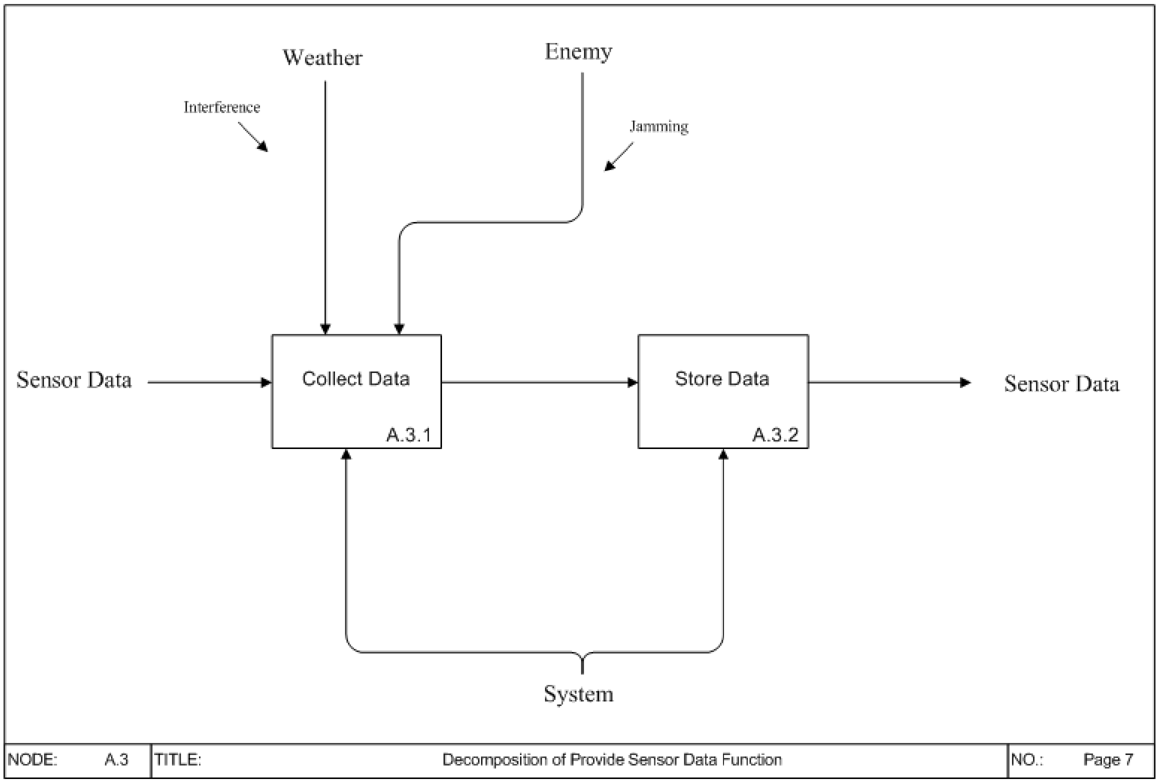

Figure 17. Decomposition of "Provide Sensor Data" function.

The "Collect Data" function can be further decomposed into the subfunctions "Receive Sensor Data" and "Perform Diagnostic Self-Test."

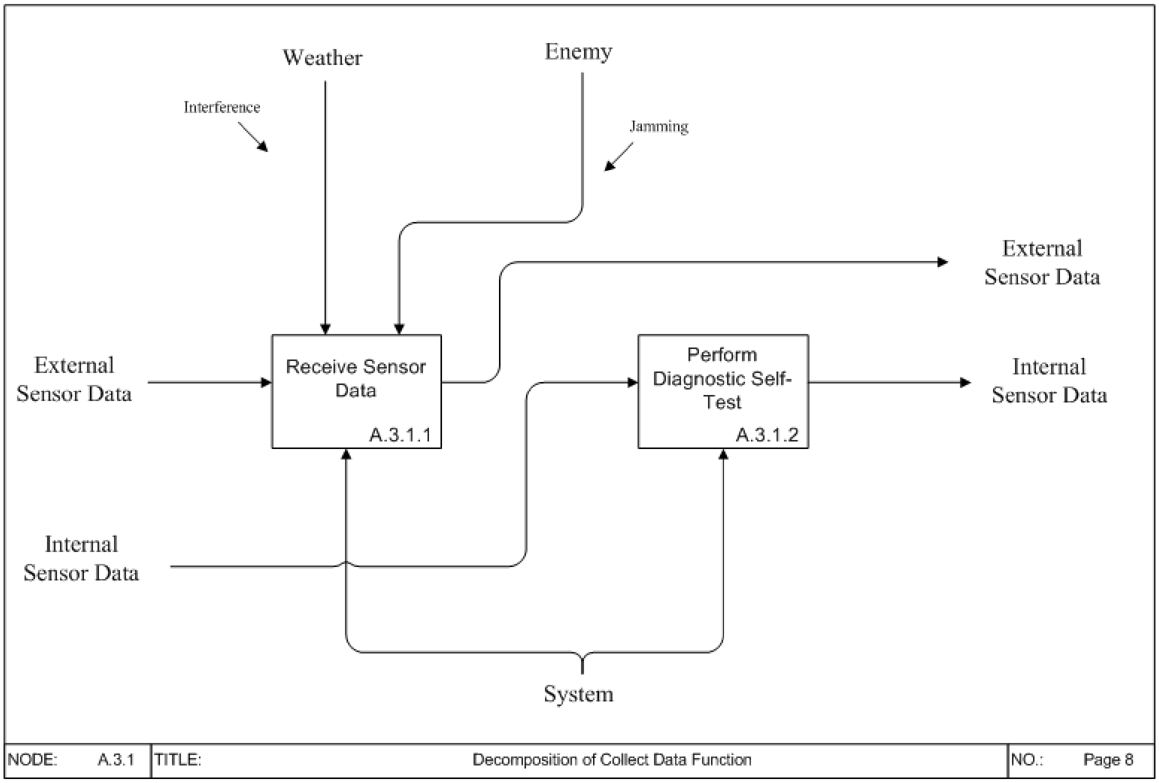

Figure 18. Decomposition of "Collect Data" function.

In summary, a Systems Engineering approach was used to design the generic solution. A DRM was established and was used to produce the OV1. The DRM was also used, along with the IDEF0 model to produce the External Systems Diagram, generic requirements, and the Functional Hierarchy and Functional Architecture. The Systems Engineering will be continued in Chapter V with the proof of concept prototype.

V. PROOF OF CONCEPT PROTOTYPE DEVELOPMENT

The bottom of the SE "Vee" model covers development and testing of components of the solution system. The DRM outlined in Chapter III, and the generic solution system architecture developed in Chapter IV, were applied to a small, inexpensive robot in an attempt to demonstrate that small footprint autonomous algorithms could be developed and implemented on a robot with low processing power. This chapter will cover the process that was followed in attempting to develop the proof of concept prototype.

A. THE SURVEYOR RENEGADE SRV ROBOT

The Surveyor Corporation is a large supplier of robots for research and education [19]. The Network-Centric Systems Engineering (NCSE) Track and Lab, in the Systems Engineering department at NPS, commissioned the design and construction of a scaled-up off-road version of the SRV-1 robot. The SRV-1 is widely used for research and education and utilizes open source software and firmware [18].

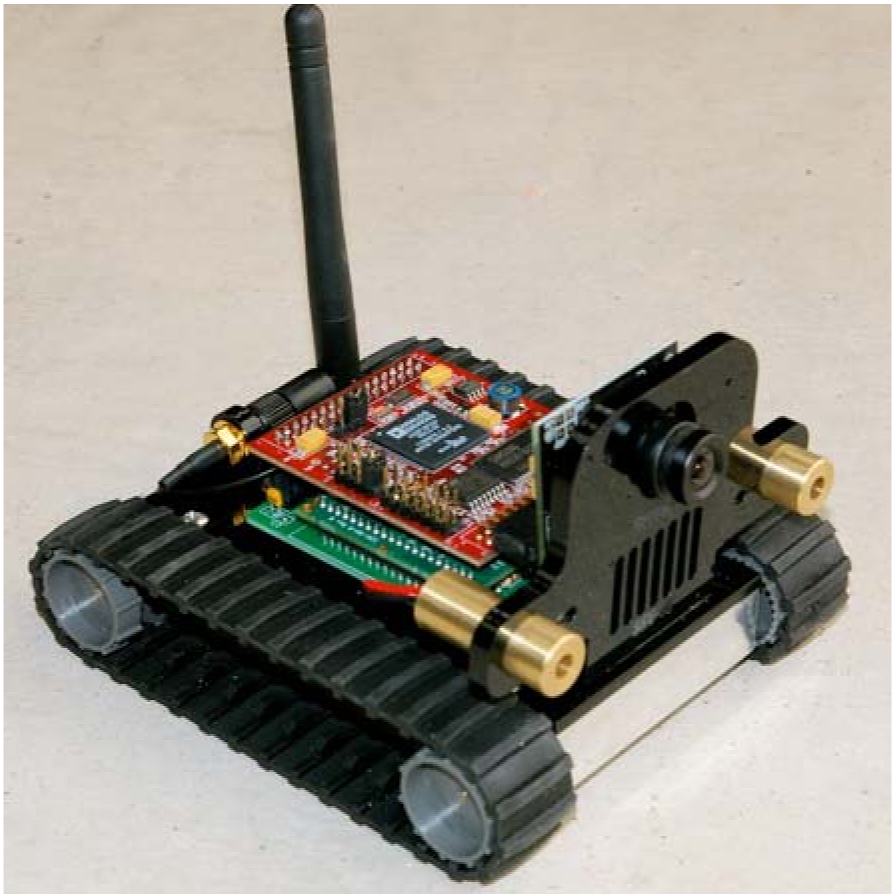

Figure 19. The Original SRV-1 robot [From 20].

Students and faculty in the NPS Systems Engineering department wanted a robot that could be integrated into a wireless smart sensor network. The Surveyor Corporation Partnered up with Inertial Labs to design the Renegade SRV robot to meet the needs of the NPS NCSE Track, of the Systems Engineering Department. Surveyor Corp. and Inertia Labs utilized elements from the SRV-1, including the 1.3 megapixel camera and Blackfin processor to design the Renegade SRV [20].

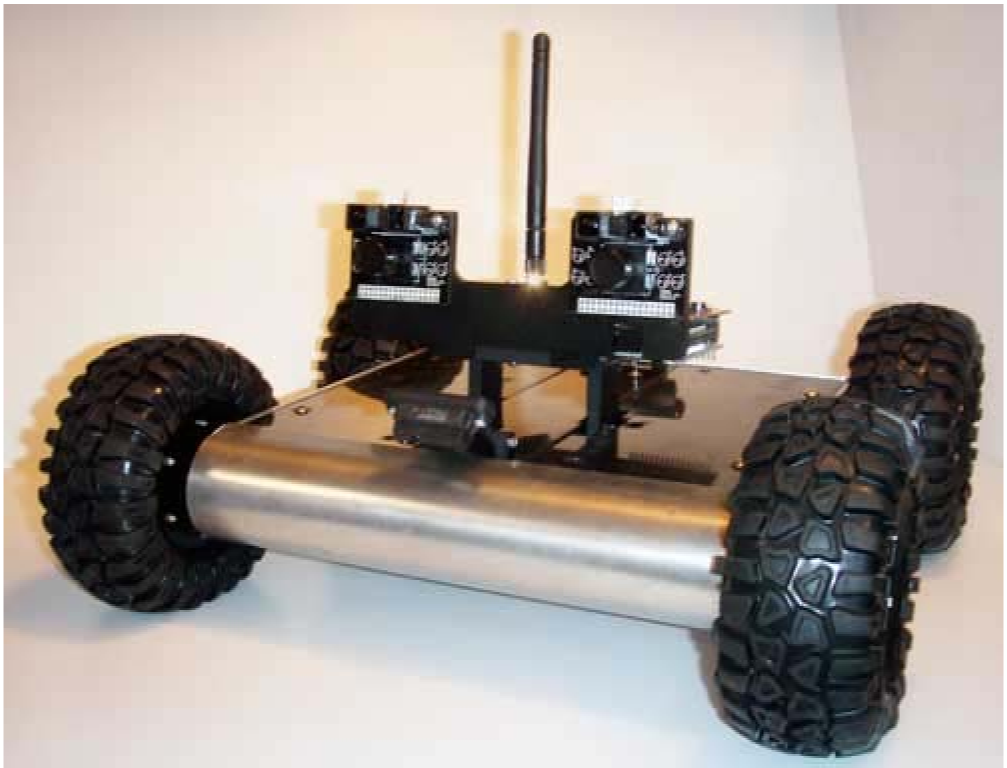

Figure 20. The Renegade SRV robot [From 21].

The Renegade SRV robot is built by Inertia-Labs and is a scaled up version of the older SRV model and contains open source electronics from the Surveyor Corporation. It has an aluminum chassis and four “rock crawler” tires designed for moderately rough terrain. The speed controller of the SRV can receive an input from each encoder. The Renegade model used for this research contains two Blackfin CPU units in a master/slave configuration. It contains two 1.3 MP video cameras on a tiltable platform, a GPS receiver and two IR range sensors (one facing forward, one facing aft) [21]. It also has an 802.11b/g wireless network adapter. Each wheel is attached to a planetary gear motor and encoder [21]. The Renegade SRV also has a speed controller that can accept inputs from each encoder and relay the encoder data to the Blackfin CPUs [21].

The Renegade can be controlled by a number of methods. It can be controlled directly by the use of individual commands. These commands can be relayed to the SRV through a console or web browser program while connected via 802.11b/g. It also has a proprietary C interpreter that interprets code in C syntax.

The Renegade was not designed and built specifically for the research associated with this thesis. It was selected for the proof of concept prototype after the NPS NCSE Track of the SE department had acquired a short initial production run of 12 robots. It was selected specifically because of its small size and low cost. It is inexpensive compared to other robots available for experimentation (each unit costs about $1200 [21]).

B. PROPOSED SYSTEM CONCEPT

To simulate a solution to the problem stated above, the Renegade SRV would have to perform a number of specific tasks and meet certain metrics. These tasks and metrics were based on the metrics and requirements outlined in the DRM in Chapter III. Due to time constraints, the scope of the metrics used in developing the proof of concept prototype was scaled to focus only on autonomous navigation capabilities.

1. Tasks

In order to be considered a successful proof of concept prototype, the system must be able to perform the following tasks:

- Communicate Wirelessly with the user.
- Provide status reports to the User.
- Maneuver to an ordered position.
- Stand by for follow-on tasking.

These tasks were derived from the requirements outlined in the DRM, as well as the functions outlined in the functional architecture of the Solution System Architecture presented in Chapter IV.

2. Metrics

The right side of the SE “Vee” Model involves integration and testing of system components. Testing is conducted to ensure that the system meets some validation criteria. These criteria are outlined in a number of metrics. The following metrics are derived form the metrics outlined in the DRM:

Metric #	Metric Type	Description of Metric	Specification
M1	Seconds	Time to maneuver to location	1 second for every meter of distance
M2	Millimeters	Distance from ordered location (error)	10mm/m or 1% error

Table 4. Metrics for the Proof of Concept prototype.

The tasks and metrics stated above were the guidelines for the development of the control code used in the Renegade SRV. Their application toward system validation will be described in more detail later in this chapter.

C. DIFFERENTIAL DRIVE THEORY

Though the Renegade SRV has GPS, accelerometers, IR sensors, and a Compass, these components were not used in the development of the proof of concept prototype. They were all determined very early in experimentation to be unreliable at best. The navigation boards on several of the robots used were either damaged or could not get accurate readings (if any) from these components. The only components that consistently gave accurate readings were the encoders attached to each wheel. For this reason, the decision was made to focus on using readings from the encoders to determine how much distance the robot had travelled through the use of odometry based navigation.

The Renegade SRV is a Differential Drive robot. This means that the wheels on the left side of the robot move independently of the wheels on the right side. The differential drive concept is illustrated in the following diagram:

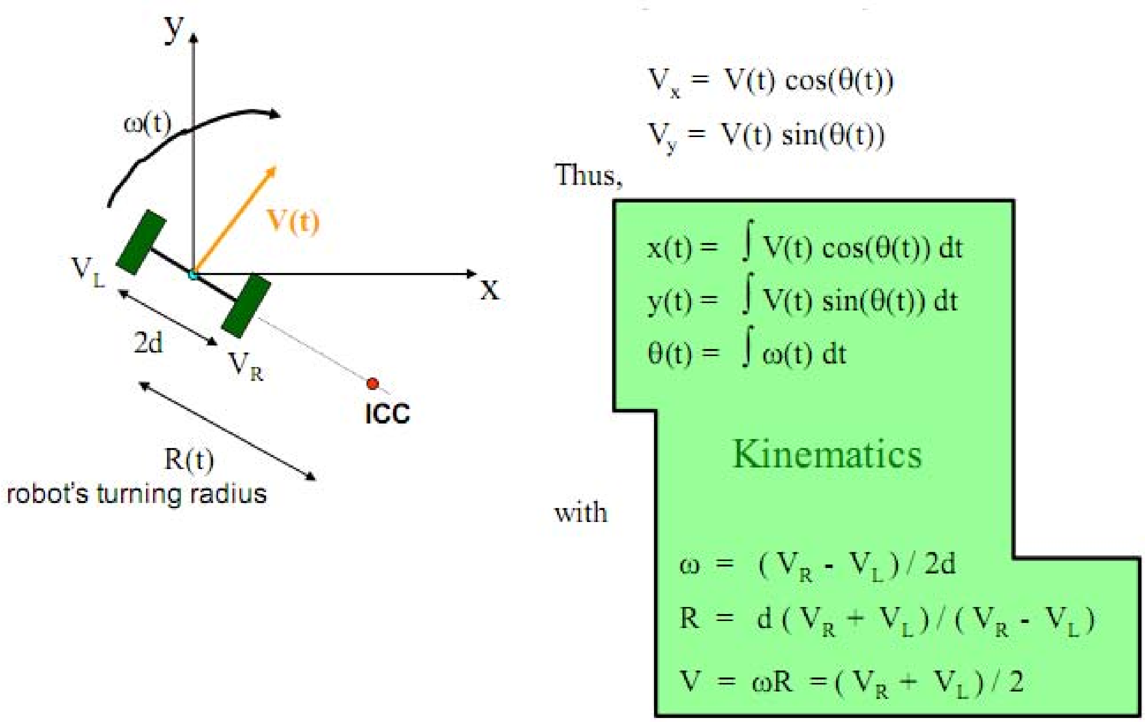

Figure 21. Differential Drive Kinematics [After B22].

Figure 21 demonstrates a robot moving through 2-dimensional (2-D) Cartesian space. Unless the velocity of the wheels on the left side of the robot are precisely equal to the velocity of the wheels on the right side, the motion of the robot is curvilinear. The top set of equations in the green box on the right represent the robot's "pose." Pose is made up of the combination of the robot's location (made up of x(t) and y(t) in 2-D space) and the robot's orientation (represented by θ(t)). These formulas were used in developing the control code for the robot. The control code implemented in the Renegade SRV is presented in Appendix A.

D. DEVELOPMENT AND IMPLEMENTATION OF THE ODOMETRY MODEL

The formulas presented in the previous section were implemented in a control code to attempt to track the motion of the robot in 2-D Cartesian space. The goal was to have the SRV keep track of its movement in order to report its current position in order to calculate the bearing, range, and motor power required to maneuver to the ordered position. This would be used whether the robot was executing current tasking or standing by for follow-on tasking for the operator. The differential drive concept was applied to the robot through the use of odometry based navigation.

Odometry based navigation involves the use of the encoders attached to the robot's motors. The encoders register "ticks" of the motors as they turn. These ticks can be measured by the CPU and can be compared to previous values to determine how much the wheels have turned since a previous time. If the number of ticks per meter for the given motors is known, the number of ticks over a given interval can be used to calculate the distance travelled by the robot. Experiments were conducted to determine the number of ticks the encoders register for a set distance travelled. The proof of concept software developed in this thesis is presented in Appendix A. The detailed results of these tests are presented in Appendix B. Based on experimentation, the robots tested an average 1755 encoder ticks per meter of distance travelled by the wheel. This knowledge as well as the length of the wheelbase of the robots, which was measured as 31.5 cm, was combined with the differential drive equations to calculate the motion and pose of the robot in 2-D space. The motion and pose of the robot in 2-D space was needed to create and implement autonomous navigation protocols that would maneuver the robot in such a way as to execute tasking.

Odometry based navigation is an ideal model of robot motion. It assumes no slip in the wheels of the robot. For this reason, it tends to have a large amount of error in execution. This is explored further later in the analysis of the test results.

E. VALIDATION TESTING FOR THE ODOMETRY BASED MOTION TRACKING ALGORITHM

The center of the right side of the SE "Vee" Model represents validation testing for specific components of the system prior to integration. The odometry based motion tracking algorithm developed represents only a small part of the solution system. It is, however, critical to the functioning of the required system and the focus of this thesis.

The odometry based algorithm for tracking the motion of the SRV was implemented and tested extensively. For metric M1, the first metric in table 4, the SRV met the specification of at least 1 m/s of transit velocity. The average velocity for the 30 trial runs was 1.23 m/s. Unfortunately, the system did not meet the specifications for the second metric, M2, of no more than 1% error. Over the 30 trial runs, the average error percentage for motion tracking in the x direction was 41%. The error for the y direction was 117%. The error for theta was 89%. Since the error was so high, the method utilized was deemed unsuitable for navigation. Due to time constraints, development of different methods of tracking motion was not attempted.

Though the specification of the first of two metrics was met, the second metric was by far the more important of the two in the development of the proof of concept prototype. The full set of test results is presented in Appendix C, and the results are analyzed in Chapter VI.

In summary, this chapter covered the process followed to apply the DRM and generic system architecture to the proof of concept prototype. It covered the concepts of differential drive and odometry based navigation and the tests conducted on the SRV to determine how well it met the specifications outlined for the system. Future research that could scale from this proof of concept system is also discussed in the next chapter.

VI. CONCLUSIONS AND RECOMMENDATIONS

This thesis outlines the process followed to develop a system that solves the problem of high manpower requirements for intelligence collection, high cost of automated systems, and risk to personnel when a non-expendable MSP is lost in a hazardous area. A DRM was developed, external systems diagram, generic requirements, functional architecture hierarchy, and functional architecture decomposition were developed for a possible generic solution, and a proof of concept prototype was developed and tested in a scoped environment.

A. THE USE OF ODOMETRY BASED MOTION TRACKING

The choice to use odometry-based motion tracking algorithms is not unrealistic. A magnetic compass, an inertial navigation system, and a GPS receiver can all be subject to intentional or unintentional interference. It is a good idea to have odometry-based motion tracking as a backup for these systems. In practice, odometry-based navigation systems are used for making a prediction of where the system is located. Since odometry represents an ideal case, it is highly subject to external sources of error, such as wheel slip. For this reason, this prediction is augmented with some sort of correction for the error. The correction could be some additional calculation based on a known error for each set of circumstances, or could be based on external sensors such as infrared, ultrasonic, or laser motion sensors. A map of the area to be explored could also be used (e.g., through computer vision algorithms on video data) and compared to the calculations made by the robot. Such a map would most likely require external sensors (e.g., cameras) to determine boundaries and obstacles in the AOI. Such correction capabilities are beyond the scope of this thesis and beyond the limitations of the firmware installed on the SRV robot.

B. FAILURE OF RENEGADE SRV TO MEET SPECIFICATIONS

The results of the tests indicate that the Renegade SRV does not meet the specifications outlined for the proof of concept prototype. Again, the SRV is a first generation, low-cost robot and through this thesis, recommendations for updates can help improve this system and be used as expendable unmanned systems. The test results suggest that this is due to two factors: Wheel slip, and computational limitations of the SRV firmware. Wheel slip is common in all wheeled vehicles [22]. It is as unavoidable as friction in any system with moving parts and can account for a great deal of error in any case utilizing odometry.

The limitations of the firmware seem to be much more responsible for the high error rate seen in the results of the validation testing for the SRV. Since the SRV firmware is not designed to handle floating point arithmetic, all calculations must be done with integers. This includes the trigonometric functions utilized in the calculation of the robot's pose. The sin and cos values are taken from a lookup table. The table multiplies the values by 1000 and gives the integer value of this amount. This constant rounding can account for a great deal of the error seen in the odometry calculations.

Though the SRV failed to meet the specifications of the proof of concept prototype, it should not be assumed that the SRV cannot meet the needs specified in the DRM or the generic system architecture. The lack of floating point arithmetic can be corrected with a different firmware load. The lack of correction for error inherent in odometry based motion tracking can be solved by adding some sort of external sensor capability or mapping correlation capability to determine the robot's most likely position. Also, the GPS and inertial navigation capabilities can be utilized to determine the robot's actual position in 2-D space.

C. RECOMMENDATIONS FOR FURTHER RESEARCH

The Renegade SRV is a great candidate for a possible system solution (low-cost, can "plug and play" sensors, open to various software, etc. Itis a great system platform to start from in the low-cost unmanned ground vehicle world). It can handle moderately rough terrain, is programmable and scalable. At this time, it is limited mostly by the firmware loaded on it. A new set of firmware that allows for floating point arithmetic and high precision trigonometric calculations could dramatically reduce the calculation error of the odometry based calculations. Universities and hobbyists are actively developing new firmware loads for the SRV Blackfin processor. One such firmware under development is based on a lightweight linux kernel [23]. These new firmware solutions should be explored to improve the capabilities of the Renegade SRV and possibly develop a solution to the problem stated in this thesis.

Beyond the Renegade SRV, there are other candidates that should be explored. The SUGV and AMCR programs presented in Chapter I are two examples of systems utilizing low-cost hardware and small footprint software. Though they are not inexpensive enough yet to be considered expendable, they offer examples of possible areas of development for a possible solution to the problem. As time goes on, the hardware necessary for these systems may go down in price enough for them to be used in a solution to the problem outlined in this thesis.

In conclusion, the process followed in this thesis should not be considered a failure. It presented a capability gap in our military. It produced an architecture for a generic solution to the problem, and produced insight into the development of such a solution system. This information will hopefully be used in the future to help develop an actual solution to the problem of lack of constant sensor coverage over large hazardous areas by an autonomous mobile sensor platform. Future applications should also include automated intelligence of monitoring, detecting, fusing, predicting and reacting to anomalous behaviors (i.e., automate intelligence officers, similar to automating the authors experience in Iraq, described in Chapter I). Automated reactions are achievable with low-cost, expendable robots, with certain rules of engagement allowed (i.e., self destroy and destroy discovered IEDs, in AOI where no humans are present). Future Network-Centric Warfare will be low cost, expendable robots, autonomous control, intelligence automation (automating intelligence experts at the robot) and new advanced sensors required for unmanned systems.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] “Chronological History; U.S. Naval Communication,” United States Navy Radioman Association. Accessed September 17, 2010, from www.radioman.org/history.pdf.

[2] “Terrorist Use of Google Earth Raises Security Fears,” National Geographic News, March 12, 2007. Accessed September 21, 2010, from http://news.nationalgeographic.com/news/2007/03/070312-google-censor.html.

[3] “Facts and Figures on U.S. Troop Drawdown in Iraq,” August 31, 2010. Accessed September 21, 2010, from http://www.america.gov/st/texttrans-english/2010/August/20100831154740su0.065777.html.

[4] “Combat Camera: US Troops in Iraq Recover Downed UAV,” *The Tension*, January 29, 2009. Accessed September 21, 2010, from http://thetension.blogspot.com/2009/01/combat-camera-us-troops-in-iraq-recover.html.

[5] “FalconView.” Accessed September 28, 2010, from http://www.falconview.org/trac/FalconView.

[6] “Military Robots / Unmanned Ground Vehicles.” Accessed September 17, 2010, from http://www.globalsecurity.org/military/systems/ground/ugv.htm.

[7] Army Brigade Combat Team Modernization, “XM1216 Small Unmanned Ground Vehicle (SUGV).” Accessed September 18, 2010, from http://www.bctmod.army.mil/systems/sugv/index.html.

[8] “XM1216 Small Unmanned Ground Vehicle (SUGV).” Photographs. Accessed September 29, 1010, from http://www.bctmod.army.mil/pic/ugv/index.html.

[9] Army Brigade Combat Team Modernization. “XM1216 Small Unmanned Ground Vehicle (SUGV) Media Flier.” Accessed September 18, 2010, from http://www.bctmod.army.mil/systems/sugv/index.html.

[10] SPAWAR. “Autonomous Mobile Communication Relays.” Accessed September 19, 2010, from http://www.spawar.navy.mil/depts/d30/d37/d371/d371_idx.html.

[11] P. Stubblefield, “Security Enhancement of Littoral Combat Ship Class Utilizing an Autonomous Mustering and Pier Monitoring System,” M.S. thesis, Naval Postgraduate School, 2010.

[12] D. Buede, *The Engineering Design of Systems: Models and Methods.* New York: John Wiley and Sons, 2000.

[13] “Universal Naval Task List,” OPNAV, January 30, 2007.

[14] “Universal Joint Task List,” Joint Staff. Accessed September 28, 2010, from www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m350004d.pdf.

[15] “Common Operational Activity List COAL v2.0,” Department of Defense, September 2007.

[16] Collection of military photographs. Accessed September 21, 2010, from http://militarystockphoto.photoshelter.com/image/I0000es85ItPnd60.

[17] “QinetiQ North America Introduces Field-Transformable Dragon Runner Small Unmanned Ground Vehicle,” [Photograph]. Accessed September 29, 2010, from http://www.tactical-life.com/online/news/qinetiq-north-america-introduces-field-transformable-dragon-runner-small-unmanned-ground-vehicle/.

[18] Photographs taken from Pacific Air Forces Web site. Accessed September 29, 2010, from http://www.pacaf.af.mil/photos/mediagallery.asp?galleryID=1073&?id=-1&page=7&count=48.

[19] “Corporate Background,” Surveyor Corporation. Accessed September 26, 2010, from http://www.surveyor.com.

[20] SRV-1 Blackfin Mobile Robot [Photograph]. Accessed September 29, 2010, from http://surveyor-corporation.stores.yahoo.net/srrowestkit.html.

[21] Renegade SRV [Photograph]. Accessed September 29, 2010, from http://www.inertialabs.com/Renegade.htm.

[22] D. Davis, “Robot Kinematics.” Class Lecture, CS4313, Naval Postgraduate School, Monterey, January 2010.

[23] “Surveyor Robotics Forum,” Surveyor Corporation. Accessed September 26, 2010, from http://www.surveyor.com.

APPENDIX A ODOMETRY BASED MOTION TRACKING CODE FOR RENEGADE SRV

```
/* code to calculate X Y and Theta of robot after
driving for a set number of seconds */


void updateEncoders(int *enc1_start,int *enc2_start,int *enc3_start,int
*enc4_start,int *xPos,int *yPos,int *theta)
{

	/* ticks per meter and wheel base */
	int TICKS = 1755;
	int D = 315;

	/* get new values for encoders */
	int enc1=encoderx(1);
	int enc2=encoderx(2);
	int enc3=encoderx(3);
	int enc4=encoderx(4);

	/* calculates encoder difference */
	int enc1Diff = enc1 - *enc1_start;
	int enc2Diff = enc2 - *enc2_start;
	int enc3Diff = enc3 - *enc3_start;
	int enc4Diff = enc4 - *enc4_start;

	printf("\n\nEncoder Difference: %d %d %d %d\n", enc1Diff, enc2Diff,
	enc3Diff, enc4Diff);

	/* calculates distance travelled by wheels on each side as average of
	each wheel */
	int rDist = (enc3Diff + enc4Diff)*100/(2*TICKS);
	int lDist = (enc1Diff + enc2Diff)*100/(2*TICKS);
	printf("\nRight Distance: %d\n", rDist);
	printf("\nLeft Distance: %d\n", lDist);

	/* calculate total distance travelled this interval as average of left
	and right distances */
	int dist = (rDist+lDist)/2;

	printf("\nTotal Distance Travelled: %9d", dist);

	/*  Calculate change in theta */
	int deltaTheta = (rDist - lDist)*10000/D;

	/*  The value of R is a coefficient used to calculate distance travelled
	*/
	int R = dist*10000/deltaTheta;

	/* converts change in theta from radians to degrees – sin and cos tables
	can only use degrees */
	deltaTheta = radToDeg(deltaTheta);
	printf("\nDelta theta in degrees: %9d\n", deltaTheta);
```

```
        /*  Calculate amount of distance travelled in the x direction */
        int xTraveled = R*(-sin(*theta)+sin(*theta+deltaTheta))/1000;
        int yTraveled = R*(cos(*theta)-cos(*theta+deltaTheta))/1000;

        printf("\nX travelled this leg: %9d", xTraveled);
        printf("\nY travelled this leg: %9d", yTraveled);

        xTraveled = xTraveled + *xPos;
        *xPos = xTraveled;

        yTraveled = yTraveled + *yPos;
        *yPos = yTraveled;

        *theta = deltaTheta + *theta;


        int power = analogx(0);
        printf("\nPower Remaining: %d\n", power);
}


int radToDeg(int rads)
{
        /* converts radians to degrees */
        return(rads*180/31416);
}


int normalizeDegs(int degs)
{
        /* accounts for wraparound at 360 degrees */
        if(degs>360)
        {
             degs = degs%360;
        }
}

/***************/
/***** main *****/
/***************/

printf("\nStarting Program\n");

/* set encoders to initial value */

int *enc1_start, start1=0;
int *enc2_start, start2=0;
int *enc3_start, start3=0;
int *enc4_start, start4=0;

enc1_start = &start1;
enc2_start = &start2;
enc3_start = &start3;
enc4_start = &start4;

*enc1_start=encoderx(1);
*enc2_start=encoderx(2);
```

```
*enc3_start=encoderx(3);
*enc4_start=encoderx(4);

/* cartesian location based on odometry */
int *xPos, xStart =0;
int *yPos, yStart =0;
int *theta, tStart =0;

xPos  = &xStart;
yPos  = &yStart;
theta = &tStart;

printf("\nStarting Pos: x %d y %d theta %d", *xPos, *yPos, *theta);

delay(500);

/*  sets motor power – 100 on the left and 100 on the right motors */
motorx(100, 100);
delay(1500);
motorx(0, 0);
delay(500);

/*  calls the update encoders function to calculate how much the robot has
moved since the last time encoders were measured */
updateEncoders(enc1_start, enc2_start, enc3_start, enc4_start, xPos, yPos,
theta);

printf("\nEnd Position: x %d y %d theta %d", *xPos, *yPos, *theta);
printf("\n");
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B TEST RESULTS FOR ENCODER CALIBRATION

The Renegade was driven in a straight line for a set period of time. The distance covered and the number of ticks registered by each encoder over the distance covered were recorded. These were used to calculate the number of ticks per meter registered by each encoder. The distances traveled were measured in meters. Each mean (μ) encoder value represents ticks per meter of distance travelled.

	μ Enc1	**M Enc2**	**μ Enc3**	**M Enc4**	**μ all**	**(ticks/m)**			
	1,761	1,753	1,756	1,749	1,755				
Run	Enc1 (ticks)	Enc2 (ticks)	Enc3 (ticks)	Enc4 (ticks)	Dist (m)	Enc1 (ticks/m)	Enc2 (ticks/m)	Enc3 (ticks/m)	Enc4 (ticks/m)
1	7108	7081	7081	7060	4.04	1,760	1,754	1,754	1,748
2	7140	7114	7121	7098	4.06	1,759	1,753	1,755	1,749
3	7177	7142	7158	7134	4.07	1,763	1,754	1,758	1,752
4	7119	7082	7105	7080	4.04	1,760	1,751	1,757	1,751
5	7119	7088	7092	7076	4.04	1,762	1,754	1,755	1,751
6	7292	7262	7274	7253	4.12	1,768	1,761	1,764	1,759
7	7294	7254	7270	7246	4.13	1,767	1,757	1,761	1,755
8	7227	7186	7213	7190	4.11	1,757	1,747	1,753	1,748
9	7260	7224	7244	7222	4.13	1,758	1,749	1,754	1,749
10	7238	7198	7214	7194	4.12	1,759	1,749	1,753	1,748
11	10152	10100	10126	10086	5.77	1,760	1,751	1,756	1,749
12	10156	10095	10138	10102	5.78	1,756	1,745	1,753	1,747
13	10162	10104	10146	10114	5.79	1,756	1,746	1,754	1,748
14	10184	10132	10160	10124	5.79	1,760	1,751	1,755	1,749
15	10153	10094	10128	10090	5.78	1,758	1,748	1,753	1,747
16	19207	19122	19124	19012	10.88	1,765	1,758	1,758	1,747
17	19212	19118	19128	19008	10.90	1,762	1,753	1,754	1,743
18	19101	19018	19026	18902	10.82	1,765	1,757	1,758	1,746
19	19404	19320	19304	19180	11.00	1,765	1,757	1,756	1,744
20	19350	19269	19268	19144	10.98	1,763	1,756	1,755	1,744

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C TESTS RESULTS FOR MOTION TRACKING CODE

	Orders to SRV			Pose Calculated by SRV			Actual Pose			% Error			
Run	L motor (%)	R motor (%)	time (ms)	x	Y	θ	x	y	θ	x	y	θ	m/s
1	100	100	750	0.97	0	0	1.22	-0.05	-7	0.20	1.00	1.00	1.63
2	100	100	750	0.97	0	0	1.21	-0.03	-3	0.20	1.00	1.00	1.61
3	100	100	750	0.96	0	0	1.21	-0.01	-2	0.21	1.00	1.00	1.61
4	100	75	750	0.28	0.01	1	0.85	-0.08	-6	0.67	1.13	1.17	1.14
5	100	75	750	0.27	0.01	1	0.86	-0.06	-6	0.69	1.17	1.17	1.15
6	100	75	750	0.28	0.01	1	0.86	-0.08	-4	0.67	1.13	1.25	1.15
7	75	100	750	0.54	0.02	5	0.95	0.12	8	0.43	0.83	0.38	1.28
8	75	100	750	0.55	0.02	4	0.96	0.12	4	0.43	0.83	0.00	1.29
9	75	100	750	0.52	0.02	4	0.93	0.12	5	0.44	0.83	0.20	1.25
10	100	50	750	0.48	0	0	0.69	-0.12	-10	0.30	1.00	1.00	0.93
11	100	50	750	0.48	0	0	0.72	0.11	-10	0.33	1.00	1.00	0.97
12	100	50	750	0.5	0	0	0.72	-0.1	-7	0.31	1.00	1.00	0.97
13	50	100	750	0.48	0.02	4	0.79	0.15	10	0.39	0.87	0.60	1.07
14	50	100	750	0.4	0.01	4	0.8	0.18	10	0.50	0.94	0.60	1.09
15	50	100	750	0.44	0.01	4	0.81	0.15	5	0.46	0.93	0.20	1.10
16	100	100	1500	1.16	0.05	5	2.05	0.03	-2	0.43	0.67	3.50	1.37
17	100	100	1500	1.15	0.06	6	2.05	0.02	3	0.44	2.00	1.00	1.37
18	100	100	1500	1.22	0.08	7	2.03	0.01	3	0.40	7.00	1.33	1.35
19	100	75	1500	0.92	0.02	2	1.79	0.29	-15	0.49	0.93	1.13	1.21
20	100	75	1500	0.92	0.02	2	1.8	0.29	-13	0.49	0.93	1.15	1.22
21	100	75	1500	0.85	0.02	2	1.79	0.27	-13	0.53	0.93	1.15	1.21
22	75	100	1500	1.08	0.06	6	1.85	0.35	12	0.42	0.83	0.50	1.26
23	75	100	1500	1.22	0.08	7	1.96	0.31	13	0.38	0.74	0.46	1.32
24	75	100	1500	1.08	0.06	6	1.95	0.36	14	0.45	0.83	0.57	1.32
25	100	50	1500	1.1	0	0	1.52	0.39	-14	0.28	1.00	1.00	1.05
26	100	50	1500	1.09	0	0	1.58	0.4	-16	0.31	1.00	1.00	1.09
27	100	50	1500	1.08	0	0	1.57	0.36	-15	0.31	1.00	1.00	1.07
28	50	100	1500	1.05	0.08	9	1.75	0.55	18	0.40	0.85	0.50	1.22
29	50	100	1500	1.07	0.07	8	1.74	0.79	14	0.39	0.91	0.43	1.27
30	50	100	1500	1.02	0.07	8	1.76	0.52	13	0.42	0.87	0.38	1.22
								Mean		0.412	1.172	0.889	1.226

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Program Executive Officer, Information Warfare Systems
   Naval Yard
   Washington, D.C.